

(translation of the front page of the priority  
document of Japanese Patent Application No. 5-001686)

PATENT OFFICE  
JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of  
the following application as filed with this Office.

Date of Application: January 8, 1993

Application Number : Patent Application 5-001686

Applicant(s) : Canon Kabushiki Kaisha

December 17, 1993

Commissioner,  
Patent Office

Wataru ASOU

Certification Number 05-3028275

CFM 459 43 807  
SH 08/155,656

日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application:

1993年 1月 8日

出 願 番 号  
Application Number:

平成 5年特許願第001686号

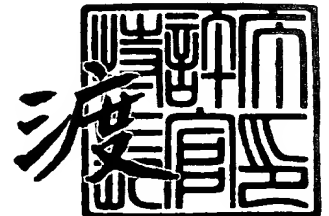
出 願 人  
Applicant(s):

キヤノン株式会社

1993年12月17日

特 許 庁 長 官  
Commissioner,  
Patent Office

麻 生 渡



出証番号 出証特平05-3028275

【書類名】 特許願

【整理番号】 2351116

【提出日】 平成 5年 1月 8日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 3/00

【発明の名称】 輪郭データ記憶方法及び装置

【請求項の数】 20

【発明者】

    【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

    【氏名】 来見田 恒昭

【特許出願人】

    【識別番号】 000001007

    【氏名又は名称】 キヤノン株式会社

    【代表者】 山路 敬三

【代理人】

    【識別番号】 100076428

    【弁理士】

    【氏名又は名称】 大塚 康德

    【電話番号】 03-3508-1864

【代理人】

    【識別番号】 100093908

    【弁理士】

    【氏名又は名称】 松本 研一

    【電話番号】 03-3508-1864

【手数料の表示】

    【納付方法】 予納

    【予納台帳番号】 003458

    【納付金額】 14,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9004561

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 輪郭データ記憶方法及び装置

【特許請求の範囲】

【請求項1】 パターンの形状をその輪郭を表す輪郭データで記憶する輪郭データ記憶装置であって、

少なくとも各輪郭点の位置情報とウエイト情報とを有する特定のウエイトに対応する輪郭データを記憶媒体に記憶する第1の記憶手段と、

ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を前記特定のウエイトに対応する輪郭データに基づいて生成する生成手段と、

前記生成手段により生成された移動情報を各輪郭点の位置情報とともに記憶媒体に記憶する第2の記憶手段と、

を備えることを特徴とする輪郭データ記憶装置。

【請求項2】 前記第1の記憶手段は、少なくとも各輪郭点の位置情報とウエイト情報とを有する特定のウエイトに対応する輪郭データを2種類のウエイトについて記憶媒体上に記憶し、

前記生成手段は、前記第1の記憶手段により記憶された2種類のウエイトの輪郭点の位置情報及びウエイト情報から、ウエイトをパラメータとする1次関数を得るためのデータを移動情報として生成する、

ことを特徴とする請求項1に記載の輪郭データ記憶装置。

【請求項3】 前記第1の記憶手段は、少なくとも各輪郭点の位置情報とウエイト情報とを有する特定のウエイトに対応する輪郭データを $n$ 種類のウエイトについて記憶媒体上に記憶し、

前記生成手段は、前記第1の記憶手段により記憶された $n$ 種類のウエイトの輪郭点の位置情報及びウエイト情報から、ウエイトをパラメータとする $n-1$ 次関数を得るためのデータを移動情報として生成する、

ことを特徴とする請求項1に記載の輪郭データ記憶装置。

【請求項4】 前記第2の記憶手段は、前記生成手段により生成された移動情報を各輪郭点の位置情報とともに記憶媒体に記憶するとともに、前記生成手段により生成された移動情報により得られる関数の次数を記憶媒体に記憶する、

ことを特徴とする請求項3に記載の輪郭データ記憶装置。

【請求項5】 前記第1の記憶手段における特定のウエイトに対応する輪郭データに基づいて各輪郭点がウエイトの変化に応じて移動可能な範囲を判定する判定手段を更に備え、

前記生成手段は、ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を、前記特定のウエイトに対応する輪郭データと前記判定手段により判定された各輪郭点の移動可能な範囲とに基づいて生成する、

ことを特徴とする請求項1に記載の輪郭データ記憶装置。

【請求項6】 前記第1の記憶手段における特定のウエイトに対応する輪郭データに基づいて各輪郭点と対をなす輪郭点を抽出する抽出手段を更に備え、

前記生成手段は、ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を、前記特定のウエイトに対応する輪郭データと前記判定手段により判定された各輪郭点の移動可能な範囲及び当該輪郭点と対をなす輪郭点の移動可能な範囲に基づいて生成する、

ことを特徴とする請求項1に記載の輪郭データ記憶装置。

【請求項7】 輪郭点の各移動可能範囲に対応した移動情報を登録移動情報として記憶媒体に記憶する第3の記憶手段を更に備え、

前記生成手段は、ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を、前記判定手段により判定された各輪郭点の移動可能な範囲に応じて前記登録移動情報を検索して獲得する、

ことを特徴とする請求項5に記載の輪郭データ記憶装置。

【請求項8】 輪郭点及びその輪郭点と対をなす輪郭点の移動可能範囲に対応した移動情報を登録移動情報として記憶媒体に記憶する第3の記憶手段を更に備え、

前記生成手段は、ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を、前記判定手段により判定された各輪郭点の移動範囲と当該輪郭点と対をなす輪郭点の移動可能範囲により前記登録移動情報を検索して獲得する、

ことを特徴とする請求項6に記載の輪郭データ記憶装置。

【請求項9】 前記第3の記憶手段は、輪郭点の移動可能な範囲をランク分

けして各ランク毎の移動情報により登録移動情報を構成し、前記ランク分けの細かさを異にする複数の登録情報を記憶媒体に記憶し、

前記複数の登録移動情報より1つの登録情報を指定する指定手段を更に備え、

前記生成手段は、ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を、前記判定手段により判定された各輪郭点の移動可能な範囲に応じて前記指定手段により指定された登録移動情報を検索して獲得する、

ことを特徴とする請求項7に記載の輪郭データ記憶装置。

【請求項10】 前記第3の記憶手段は、輪郭点の移動可能な範囲とその輪郭点と対をなす輪郭点の移動可能な範囲により決定される輪郭点の移動可能量に基づいてランク分けして各ランク毎の移動情報により登録移動情報を構成し、前記ランク分けの細かさを異にする複数の登録情報を記憶媒体に記憶し、

前記複数の登録移動情報より1つの登録情報を指定する指定手段を更に備え、

前記生成手段は、ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を、前記判定手段により判定された各輪郭点の移動範囲と当該輪郭点と対をなす輪郭点の移動可能範囲により前記指定手段にて指定された登録移動情報を検索して獲得する、

ことを特徴とする請求項8に記載の輪郭データ記憶装置。

【請求項11】 パターンの形状をその輪郭を表す輪郭データで記憶する輪郭データ記憶方法であって、

少なくとも各輪郭点の位置情報とウエイト情報とを有する特定のウエイトに対応する輪郭データを記憶媒体に記憶する第1の記憶工程と、

ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を前記特定のウエイトに対応する輪郭データに基づいて生成する生成工程と、

前記生成工程により生成された移動情報を各輪郭点の位置情報とともに記憶媒体に記憶する第2の記憶工程と、

を備えることを特徴とする輪郭データ記憶方法。

【請求項12】 前記第1の記憶工程は、少なくとも各輪郭点の位置情報とウエイト情報とを有する特定のウエイトに対応する輪郭データを2種類のウエイトについて記憶媒体上に記憶し、

前記生成工程は、前記第1の記憶工程により記憶された2種類のウェイトの輪郭点の位置情報及びウェイト情報から、ウェイトをパラメータとする1次関数を得るためのデータを移動情報として生成する、

ことを特徴とする請求項11に記載の輪郭データ記憶方法。

【請求項13】 前記第1の記憶工程は、少なくとも各輪郭点の位置情報とウェイト情報とを有する特定のウェイトに対応する輪郭データをn種類のウェイトについて記憶媒体上に記憶し、

前記生成工程は、前記第1の記憶工程により記憶されたn種類のウェイトの輪郭点の位置情報及びウェイト情報から、ウェイトをパラメータとするn-1次関数を得るためのデータを移動情報として生成する、

ことを特徴とする請求項11に記載の輪郭データ記憶方法。

【請求項14】 前記第2の記憶工程は、前記生成工程により生成された移動情報を各輪郭点の位置情報とともに記憶媒体に記憶するとともに、前記生成工程により生成された移動情報により得られる関数の次数を記憶媒体に記憶する、

ことを特徴とする請求項13に記載の輪郭データ記憶方法。

【請求項15】 前記第1の記憶工程における特定のウェイトに対応する輪郭データに基づいて各輪郭点がウェイトの変化に応じて移動可能な範囲を判定する判定工程を更に備え、

前記生成工程は、ウェイトの変化に対応して輪郭点の位置を移動させるための移動情報を、前記特定のウェイトに対応する輪郭データと前記判定工程により判定された各輪郭点の移動可能な範囲とに基づいて生成する、

ことを特徴とする請求項11に記載の輪郭データ記憶方法。

【請求項16】 前記第1の記憶工程における特定のウェイトに対応する輪郭データに基づいて各輪郭点と対をなす輪郭点を抽出する抽出工程を更に備え、

前記生成工程は、ウェイトの変化に対応して輪郭点の位置を移動させるための移動情報を、前記特定のウェイトに対応する輪郭データと前記判定工程により判定された各輪郭点の移動可能な範囲及び当該輪郭点と対をなす輪郭点の移動可能な範囲に基づいて生成する、

ことを特徴とする請求項11に記載の輪郭データ記憶方法。



【請求項17】 輪郭点の各移動可能範囲に対応した移動情報を登録移動情報として記憶媒体に記憶する第3の記憶工程を更に備え、

前記生成工程は、ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を、前記判定工程により判定された各輪郭点の移動可能な範囲に応じて前記登録移動情報を検索して獲得する、

ことを特徴とする請求項15に記載の輪郭データ記憶方法。

【請求項18】 輪郭点及びその輪郭点と対をなす輪郭点の移動可能範囲に対応した移動情報を登録移動情報として記憶媒体に記憶する第3の記憶工程を更に備え、

前記生成工程は、ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を、前記判定工程により判定された各輪郭点の移動範囲と当該輪郭点と対をなす輪郭点の移動可能範囲により前記登録移動情報を検索して獲得する、

ことを特徴とする請求項16に記載の輪郭データ記憶方法。

【請求項19】 前記第3の記憶工程は、輪郭点の移動可能な範囲をランク分けして各ランク毎の移動情報により登録移動情報を構成し、前記ランク分けの細かさを異にする複数の登録情報を記憶媒体に記憶し、

前記複数の登録移動情報より1つの登録情報を指定する指定工程を更に備え、

前記生成工程は、ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を、前記判定工程により判定された各輪郭点の移動可能な範囲に応じて前記指定工程により指定された登録移動情報を検索して獲得する、

ことを特徴とする請求項17に記載の輪郭データ記憶方法。

【請求項20】 前記第3の記憶工程は、輪郭点の移動可能な範囲とその輪郭点と対をなす輪郭点の移動可能な範囲により決定される輪郭点の移動可能量に基づいてランク分けして各ランク毎の移動情報により登録移動情報を構成し、前記ランク分けの細かさを異にする複数の登録情報を記憶媒体に記憶し、

前記複数の登録移動情報より1つの登録情報を指定する指定工程を更に備え、

前記生成工程は、ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を、前記判定工程により判定された各輪郭点の移動範囲と当該輪郭点と対をなす輪郭点の移動可能範囲により前記指定工程にて指定された登録移動情報

を検索して獲得する、

ことを特徴とする請求項18に記載の輪郭データ記憶方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】

本発明は文字等の印刷或いは表示等を行うための輪郭形成に用いる輪郭データを記憶する輪郭データ記憶方法及び装置に関するものである。

【0002】

【従来の技術】

従来、アウトラインフォントは直線と曲線によって表わされる文字等の輪郭データから構成されている。この曲線には3次のベジェ曲線やスプライン曲線、円弧等が一般的に用いられている。

【0003】

この輪郭データ（アウトラインデータ）は、ある特定の太さ（以後ウエイトと称する）のフォントのデータを表わすものである。即ち、細いフォント（ウエイトがライトのフォント）は細いフォント用のアウトラインデータによって表わされ、太いフォント（ウエイトがボールドのフォント）は太いフォント用のアウトラインデータによって個々に表わされている。

【0004】

【発明が解決しようとする課題】

従って、上記従来例では、複数の太さのフォントによる文字などを表示、もしくは印刷するためには、複数のウエイトのアウトラインフォントを個々に持たなければならないという問題があった。

【0005】

また、上記の問題を解決するシステムとして、太いアウトラインフォントと細いアウトラインフォントから中間のアウトラインフォントを作ることができるものもあるが、任意の太さのアウトラインフォントをつくることはできなかった。即ち、従来の輪郭点データには、任意のウエイトに対応するための各輪郭点の移動に関する情報が含まれていないために、任意のウエイトに対して各輪郭点

の位置を獲得することができなかった。

【0006】

本発明は上記の問題点に鑑みてなされたものであり、1種類のウエイトの輪郭点データで複数のウエイトの文字を発生させることを可能とするための各輪郭点毎の移動情報を生成し、輪郭データの各輪郭点の位置情報とともに移動情報を記憶する輪郭データ記憶方法及び装置を提供することを目的とする。

【0007】

【課題を解決するための手段】

上記の目的を達成するための本発明による輪郭データ記憶装置は以下の構成を備える。即ち、

パターンの形状をその輪郭を表す輪郭データで記憶する輪郭データ記憶装置であって、

少なくとも各輪郭点の位置情報とウエイト情報とを有する特定のウエイトに対応する輪郭データを記憶媒体に記憶する第1の記憶手段と、

ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を前記特定のウエイトに対応する輪郭データに基づいて生成する生成手段と、

前記生成手段により生成された移動情報を各輪郭点の位置情報とともに記憶媒体に記憶する第2の記憶手段と、

を備える。

【0008】

また、上記の目的を達成する本発明の輪郭データ記憶方法は以下の工程を備える。即ち、

パターンの形状をその輪郭を表す輪郭データで記憶する輪郭データ記憶方法であって、

少なくとも各輪郭点の位置情報とウエイト情報とを有する特定のウエイトに対応する輪郭データを記憶媒体に記憶する第1の記憶工程と、

ウエイトの変化に対応して輪郭点の位置を移動させるための移動情報を前記特定のウエイトに対応する輪郭データに基づいて生成する生成工程と、

前記生成工程により生成された移動情報を各輪郭点の位置情報とともに記憶媒

体に記憶する第2の記憶工程と、  
を備える。

【0009】

【作用】

上記の構成により、記憶媒体に記憶されている少なくとも各輪郭点の位置情報とウエイト情報を有する輪郭データより、各輪郭点の位置をウエイトの変化に対応して移動するための移動情報を生成する。そして、生成された移動情報を各輪郭点の位置情報とともに記憶媒体に記憶する。

【0010】

【実施例】

以下に添付の図面を参照して本発明の好適な実施例について説明する。

【0011】

<実施例1>

図1は本実施例の輪郭形成装置の概略制御構成を表すブロック図である。同図において、11は入力装置であり、出力すべきパターンを特定するコード情報と、そのパターンの太さを特定するウエイト情報を受け取る。尚、本実施例では、出力すべきパターンの1例として文字を扱うものとし、パターンを特定するコードを文字コードと称することにする。12は計算装置であり、入力装置11が受け取った文字コードとウエイトから該当する文字の輪郭データを計算する。13は記憶装置であり、ROM（読み出し専用メモリ）14と、RAM（ランダムアクセスメモリ）15とを備える。ROM14は、基本文字データ等を記憶し、RAM15は、CPU12が計算のためのワークエリアとして使用したり、ROM14の基本文字データをCPU12で計算して得られた文字輪郭データを一時的に記憶したりする。16は出力装置であり、CPU12で計算した結果出来上がった文字輪郭データを出力する。

【0012】

図2は本実施例の輪郭形成装置の応用の1例として、輪郭形成装置をレーザービームプリンタに組み込んだ場合の概略制御構成を表すブロック図である。同図において、21は受信部であり、ホストコンピュータからの印刷データを受信する

。22は画像形成部であり、CPU、ROM、RAMなどを備えた回路により構成されている。23は本実施例の輪郭形成装置であり、上述の図1におけるCPU12、ROM14、RAM15は画像形成部22のものを共有する。本装置の制御はCPU12によって行われ、制御のためのプログラムはROM14に記憶してある。24は出力バッファであり、画像形成装置22によって形成された画像をビットマップの形式で一時的に保存する。25は印刷部であり、出力バッファ24の内容をもとにレーザ駆動用のパルスが発生し、紙上にトナーによる記録画像を形成する。

## 【0013】

図3は、本実施例の輪郭形成装置で用いられるフォントデータの形式の1例を図示したものである。本データはゴシック体の「田」の文字である。図3の実線で表した部分は文字のウエイトが1の時の輪郭データを表す。点線で表したベクトル（矢印）は、ウエイトを変えたときの各輪郭点の移動の軌跡を表すベクトル（輪郭点移動ベクトル）である。各輪郭点移動ベクトルはX座標とY座標がウエイトの関数として表されものである。

## 【0014】

また、図4はウエイトが10の時のゴシック体の「田」の輪郭データを表す図である。本実施例においては、文字のウエイトは1が最も細く、10が最も太いものとする。

## 【0015】

図3で表したように、本実施例における輪郭データは文字の輪郭を表す座標データと、ウエイトの変化に対する輪郭座標の移動の軌跡を表す輪郭点移動ベクトルに関するデータ（以後ベクトルデータと称する）とを備える。図3に示すように、「田」という文字の各輪郭点のベクトルデータは、ベクトルがないものと、ウエイトをパラメータとするX、Y座標のそれぞれが1次関数で表されるものとなる。

## 【0016】

図6は図3の「田」という文字のアウトラインデータの構成の1部を表す図である。同図においてフラグ1は輪郭の開始／終了を表すフラグ、フラグ2は輪郭

の属性を表すフラグ、フラグ3は輪郭点移動ベクトルのX方向の関数の次数を表すフラグ、フラグ4は輪郭点移動ベクトルのY方向の関数の次数を表すフラグである。フラグ3が0のときは輪郭点移動ベクトルのX方向の座標値が変化しないことを表し、1のときはX方向の座標値がウエイトの変化に対し1次関数的、すなわち直線的に変化することを表す。同様にフラグ4が0のときは輪郭点移動ベクトルのY方向の座標値が変化しないことを表し、1のときはY方向の座標値がウエイトの変化に対し1次関数的、すなわち直線的に変化することを表す。

【0017】

また、図6において、X座標、Y座標はそれぞれの輪郭点のウエイト1における座標を表す。また、ベクトルX座標及びベクトルY座標は、ウエイト10となったときのそれぞれの輪郭点の位置が、ウエイト1の輪郭点の座標に対する移動量で表されている。

【0018】

図7は本実施例1の輪郭形成制御の手順を表すフローチャートである。まず、ステップS701において出力すべき文字コードを受信し、ステップS702にてウエイト情報を受信する。即ち、図1の入力装置11に文字コード及びウエイト情報が入力される。ステップS703では、ステップS701で受信した文字コードに従って基本の輪郭データをROM14から読み出す。ステップS704では、読み出された輪郭データの有する輪郭の点数をNmaxに代入する。そして、ステップS705ではカウンタ変数nに1を代入する。

【0019】

ステップS706ではnがNmaxを越えたかどうかを判定し、越えていればステップS712へ、越えていなければステップS707へそれぞれ進む。ステップS707では、n番目の輪郭点データがX方向の輪郭点移動ベクトルを持っているかどうかを判定（フラグ3が0でなければ輪郭点移動ベクトルが有る）し、輪郭点移動ベクトルがあればステップS708を、なければステップS709を実行する。ステップS708では輪郭点のX座標をウエイト情報及び輪郭点移動ベクトルのベクトルデータから計算する。また、ステップ709では、n番目の輪郭点のY方向の輪郭点移動ベクトルを持っているかどうかを判定（フラグ4

が0でなければ輪郭点移動ベクトルが有る)し、輪郭点移動ベクトルがあればステップS710を、なければステップS711を実行する。ステップS710ではn番目の輪郭点のY座標をウェイト及び輪郭点移動ベクトルのベクトルデータから計算する。ステップS711ではカウンタ変数nをインクリメントし、ステップS706へ戻り、次の輪郭点についての処理を行う。また、ステップS712では変換されたデータを出力装置16より送出して本処理を終了する。

【0020】

図8は輪郭点移動ベクトルを表す1次関数の式の求め方を示す図である。本例では、図6のアウトラインデータの構成で示したように、ウェイト1の輪郭点の座標(X座標、Y座標)と輪郭点移動ベクトルの終点の座標(ベクトルX座標、ベクトルY座標により示される)とにより1次関数を表している。すなわち、輪郭点移動ベクトルベクトルの始点、終点に分かっているため、容易に定数A、Bが決定され、1次関数を求めることができる。そして、それによって任意のウェイトの文字輪郭データを求めることが可能になる。

【0021】

図9は、各輪郭点の輪郭点移動ベクトルを自動的に計算で求める1例を表す図であり、2つのウェイトの輪郭データから輪郭点移動ベクトルを取り出す例を表す。この例では、「A」のウェイト1とウェイト10における輪郭点の位置座標から輪郭点移動ベクトルを求めている。即ち、ウェイト1の各輪郭点を始点とし、ウェイト10の各輪郭点を終点とするベクトルを求める。これらの値を図6のデータ構成に従って記述すると、X座標、Y座標がウェイト1の各輪郭点の座標になり、各輪郭点移動ベクトルの終点のX座標、Y座標がウェイト10のX座標、Y座標とウェイト1のX座標、Y座標との差(ベクトルX座標、ベクトルY座標)で表される。

【0022】

もちろん、各輪郭点の輪郭点移動ベクトルは他の求め方を用いて求めても良い。また、デザイナーが輪郭点移動ベクトルを修正することも可能である。もちろん求めた輪郭点移動ベクトルが同じであれば、ベクトルの求め方はウェイトの異なる文字輪郭データの生成には影響しない。

【0023】

以上説明したように、文字輪郭上の各点にウェイトを可変にするための輪郭点移動ベクトルを持たせることにより以下のような効果がある。即ち、

1. ウェイトの変更を任意の太さに、かつリアルタイムに計算することができる
2. 2種類のウェイトの違う輪郭を持ち補完する場合に比べ、ベクトルを持たない点が存在する分、データ量が少なくて済む
3. 2種類のウェイトの違う輪郭を持ち補間する場合に比べ、対応する輪郭をその場で計算する必要がないため、スピードが速い
4. 輪郭データにベクトルを付加するためデータ管理が容易であるという効果がある。

【0024】

<実施例2>

上述の実施例1では各輪郭点の移動ベクトルが1次式で表される場合を説明した。しかしながら、より複雑な図形になると、さまざまな関数で表されるベクトルデータが出現する。ベクトルデータを表す関数は輪郭点によって1次式で表されるもの、2次式で表されるもの、多項式で表されるもの、三角関数で表されるもの、あるいはウェイトを境に関数が変わるものなどがある。最も一般的には、各輪郭点移動ベクトルは多項式で表され、それぞれの輪郭点のベクトルデータはその多項式の係数を導き出すための情報を合わせて保存する。

【0025】

従って実施例2では、輪郭点移動ベクトルを2次以上の曲線で所有する輪郭形成装置について説明する。

【0026】

図10は、図1の輪郭形成装置において、各輪郭点にウェイトを変化させるための輪郭点移動ベクトルを1次の直線または2次以上の曲線で持たせた例であり、曲線で表される輪郭点移動ベクトルの1例を表す図である。これは、各輪郭点のウェイトに対する移動の軌跡を直線または曲線で表すことを意味する。ゴシック体の「1」はウェイトが太くなるに従って各輪郭点がX方向にも広がるが、そ



の移動量はY方向の移動に対して1定の比率ではない。この場合、2次以上の曲線を用いて軌跡を表すことで本来の軌跡により近い軌跡が表せる。

#### 【0027】

図11は実施例2における輪郭データの構成を表す図である。同図において、フラグ1は輪郭の開始／終了を表すフラグ、フラグ2は輪郭の属性を表すフラグである。フラグ3は輪郭点移動の軌跡におけるX方向の関数の次数を表すフラグ、フラグ4はY方向の関数の次数を表すフラグである。フラグ3が0のときはウェイトの変化に対してX方向の座標値が変化しないことを表す。フラグ3が1のときは、ウェイトの変化に対してX方向の座標値が1次関数的、すなわち直線的に変化することを表す。また、フラグ3が2のときはウェイトの変化に対してX方向の座標値が2次関数的、すなわち曲線的に変化することを表す。同様にフラグ3が3以上の場合はより複雑な曲線の動きを表すことができる。フラグ4はY方向に関してフラグ3と同等の意味を表す。各輪郭点にはその次数に応じた点の座標が対応するウェイトの順に配列ではいつている。すなわち、1次で表せる軌跡を描く点は輪郭点以外に1点、2次であれば輪郭点以外に2点の座標が格納される。そして、各輪郭点の移動ベクトルの次数に応じて、ベクトルX座標1, 2…が格納されており、各輪郭点におけるベクトルの計算式を求めるのに使用される。

#### 【0028】

図12は、図11の形式の輪郭データから所望のウェイトの文字輪郭データを計算し出力する手順を表すフローチャートである。ここで、ステップS1207におけるドラフトモードとは、文字の品位より速度を優先する出力モードを表す。このドラフトモードにおいては、すべての輪郭点の移動軌跡を1次式で近似するため、ウェイトの最も細い（ウェイト1）輪郭上の点と、ウェイトの最も太い（ウェイト10）輪郭上の点を結ぶ直線状に、所望のウェイトに対する輪郭点があるとして計算を行う。ドラフトモードでない場合は、輪郭データに基づいた次数での計算を行い、輪郭点の座標を求める。

#### 【0029】

まず、ステップS1201及びステップS1202において、入力装置11が

出力すべき文字コード及びその太さを決めるウエイト情報を受信する。ステップ S 1 2 0 3 では、ステップ S 1 2 0 1 で受信した文字コードに従って基本の輪郭データを ROM 1 4 から読み出す。ステップ S 1 2 0 4 では、読み出された輪郭データの有する輪郭の点数を  $N_{max}$  に代入する。そして、ステップ S 1 2 0 5 ではカウンタ変数  $n$  に 1 を代入する。

【0030】

ステップ S 1 2 0 6 では  $n$  が  $N_{max}$  を越えたかどうかを判定して、越えていればステップ S 1 2 1 8 へ、越えていなければステップ S 1 2 0 7 へそれぞれ進む。ステップ S 1 2 0 7 では、画像形成装置の出力モードがドラフトモードか否かをチェックし、ドラフトモードでなければステップ S 1 2 0 8 以降の処理を実行し、ドラフトモードであればステップ S 1 2 1 3 以降の処理を実行する。

【0031】

ドラフトモードでない場合、ステップ S 1 2 0 8 にて、 $n$  番目の輪郭点に対して X 方向の輪郭点移動ベクトルの有無を判定（フラグ 3 が 0 でなければ移動ベクトルが有る）する。そして、X 方向の輪郭点移動ベクトルが存在すればステップ S 1 2 0 9 にて、輪郭点の X 座標をウエイト情報及び輪郭点移動ベクトルのベクトルデータから計算する。このときの輪郭点の移動の軌跡を表す計算式は、フラグ 3 に記憶された曲線式の次数と、ベクトル X 座標 1, 2... を用いて決定する。例えば、曲線の次数が 2 であれば、ベクトル X 座標 1 及び 2 が用いられる。同様にして、ステップ S 1 2 1 0 及びステップ S 1 2 1 2 にて、輪郭点の Y 座標が求められる。そして、 $n$  を 1 つインクリメントしてステップ S 1 2 0 6 へ戻り、次の輪郭点に対して上述の処理を繰り返す。そして、全輪郭点に対して処理が終了するとステップ S 1 2 0 6 よりステップ S 1 2 1 8 へ進み、データ出力装置 1 6 よりデータを送出して本処理を終了する。

【0032】

1 方、出力モードがドラフトモードの場合は、ステップ S 1 2 0 7 よりステップ S 1 2 1 3 へ進む。ステップ S 1 2 1 3 では、 $n$  番目の輪郭点が X 方向の輪郭点移動ベクトルを持っているか否かをチェックし、持っていればステップ S 1 2 1 4 へ進む。ステップ S 1 2 1 4 では、ウエイト 1 とウエイト 1 0 の輪郭点の X

座標より輪郭点移動ベクトルの計算式を1次式として決定し、これを用いて所望のウェイトに対する輪郭点のX座標を求める。Y座標についてもステップS1215及びステップS1216により、上述のステップS1213、1214と同様に求めることができる。そして、ステップS1217にてnを1つカウントアップし、ステップS1206へ戻る。

【0033】

図13は、各輪郭点のベクトルを自動的に計算で求める1例であり、3つ以上のウェイトの輪郭データから非直線のベクトルを取り出す様子を表す図である。この例では、「1」のウェイト1、ウェイト5とウェイト10から輪郭点移動ベクトル（2次式で表される）を求めている。即ち、ウェイト1の各輪郭点を始点とし、ウェイト5の点を通りウェイト10の各輪郭点を終了点とする2次式の輪郭点移動ベクトルを求める。この場合、前述の図11のデータ構成では、X、Y座標はウェイト1の輪郭点の座標を表し、ベクトルX、Y座標1はウェイト10の輪郭点の座標を示す（実際にはウェイト1に対する相対位置）。また、ベクトルX、Y座標2はウェイト5の輪郭点の座標を示す。更に、ベクトルX、Y座標3は、ウェイト3の輪郭点の座標を示す。

【0034】

メモリ容量やCPUのパワーに応じて、1次式のデータを持つか、2次式以上のデータを持つかを選択することも可能である。このとき、2次式以上のデータを1次式のデータに変換することは容易に行える。ただし、1次式の輪郭点移動ベクトルで近似した場合は、2次式以上の輪郭点移動ベクトルを用いた場合に比べて多少の文字品位の低下をまねくおそれがある。

【0035】

以上説明したように、文字輪郭上の各点にウェイトを可変にするためのベクトルを曲線で持たせることにより以下のような効果がある。即ち、

1. より高品位のウェイト変更が行える
2. 直線で表すことができる部分は直線で表すことにより、データ量を余り増やすことなくデータを格納することができる
3. 曲線で高品位に展開するか直線で高速に展開するかの選択ができる

4. 複数の輪郭情報を持つことに比べデータ管理が容易であるという効果がある。

【0036】

### ＜実施例3＞

次に、実施例3では、1つの輪郭点から輪郭点移動ベクトルを求め、この輪郭点移動ベクトルを用いて各ウエイトに対する輪郭点の座標を求める輪郭形成装置について説明する。尚、実施例3の輪郭形成装置の構成は実施例1と同様であるのでここではその説明は省略する。

【0037】

図14、図15、図16は、図1の輪郭形成装置において、各輪郭点に対する輪郭点移動ベクトルを求める手順を表すフローチャートである。図9で示したような2種類以上のウエイトからベクトルを求める場合に比べ、1種類のウエイトの文字のみより輪郭点移動ベクトルを求めることができる。この場合、アウトラインデータの容量が減り、管理が容易になるという利点がある。

【0038】

図14のフローチャートの手順では、ウエイト1の輪郭点の位置から、水平方向の移動に対し、水平移動不可、水平移動制限有り、水平移動制限無し of 3種類の分類を行っている。この移動の分類は対になる輪郭と、他の輪郭或いは文字枠（ボディ枠）との位置関係から導き出すことができる。即ち、ウエイトを10にしたときの標準水平線幅とウエイトが1の標準水平線幅の差分、輪郭点を水平に、対の輪郭とは反対方向に動かしたときに、その輪郭点の輪郭に接触もしくは交差するかどうか、また文字のボディ枠をはみ出すかどうか、といったことから移動の可能性をフラグにセットする。同様に垂直方向についても同様の判定を行っている。

【0039】

まずステップS1401で標準水平線幅を入力する。本例では、ウエイト1のときと10のときの標準水平線幅、標準垂直線幅を入力する。また、これら標準水平線幅、標準垂直線幅は各フォントに応じて予めROM14に格納されているものとする。次に、ステップS1402にて全輪郭点数をNに代入し、輪郭点力

ウンタ c を 1 にセットする。ステップ S 1 4 0 3 では c が N より大きいかな否かをチェックし、大きければステップ S 1 5 0 1 (図 1 5) に進む。また、c が N 以下であればステップ S 1 4 0 4 へ進み、c 番目の輪郭点を取り出す。

【0040】

ステップ S 1 4 0 5 では、取り出した c 番目の輪郭点が水平方向に移動可能かどうかを調べ、不可であればステップ S 1 4 0 8 へ進み、可能であればステップ S 1 4 0 6 へ進む。ステップ S 1 4 0 6 では水平方向の移動に制限があるかな否かをチェックし、制限があればステップ S 1 4 0 9 へ、無ければステップ S 1 4 0 7 へそれぞれ進む。ステップ S 1 4 0 7 では、輪郭点の水平方向の移動に対して制限がないので、当該輪郭点の水平移動量大を表すフラグをセットする。また、ステップ S 1 4 0 5 で、水平移動が不可と判定された場合はステップ S 1 4 0 8 にて水平移動不可を表すフラグをセットする。更に、ステップ S 1 4 0 6 で水平移動に制限がある場合はステップ S 1 4 0 9 において水平移動量小を表すフラグをセットする。

【0041】

次に、ステップ S 1 4 1 0 以降で、垂直方向についての移動量をチェックする。ステップ S 1 4 1 0 では、取り出した c 番目の輪郭点が垂直方向に移動可能かどうかを調べ、不可であればステップ S 1 4 1 3 へ進み、可能であればステップ S 1 4 1 1 へ進む。ステップ S 1 4 1 1 では垂直方向の移動に制限があるかな否かをチェックし、制限があればステップ S 1 4 1 4 へ、無ければステップ S 1 4 1 2 へそれぞれ進む。ステップ S 1 4 1 2 では、輪郭点の垂直方向の移動に対して制限がないので、当該輪郭点の垂直移動量大を表すフラグをセットする。また、ステップ S 1 4 1 0 で、垂直移動が不可と判定された場合はステップ S 1 4 1 3 にて垂直移動不可を表すフラグをセットする。更に、ステップ S 1 4 1 1 で垂直移動に制限がある場合はステップ S 1 4 1 4 において垂直移動量小を表すフラグをセットする。

【0042】

そして、ステップ S 1 4 1 5 において輪郭点カウンタ c を 1 つインクリメントしてステップ S 1 4 0 3 へ戻る。このようにして、全輪郭点に対して移動量を表

すフラグがセットされると、ステップS1403よりステップS1501へ制御が進む。

【0043】

このようにして求めた全輪郭点の移動量を表すフラグから、各輪郭点の移動量を求める。図15ではこの移動量を求める手続きを示している。移動量の計算では通常、輪郭点の対となる輪郭点とあわせて必要となる太さになるよう、ベクトルを計算する。このため、移動量を計算する点のフラグと、対の輪郭点のフラグを合わせて移動量計算を行うサブルーチンをコールする。

【0044】

図15のフローチャートについて説明する。まず、ステップS1501にて、全輪郭点数をNに代入し、輪郭点カウンタcを1にセットする。ステップS1502では、cがNより大きいかな否かをチェックし、大きければ本処理を終了し、cがN以下であればステップS1503へ進む。

【0045】

ステップS1503では、c番目の輪郭点を取り出し、ステップS1504ではc番目の輪郭点と対をなす輪郭点を取り出す。この対をなす輪郭点はc番目の輪郭点に対して垂直方向及び水平方向の幅を決定する輪郭点であり、1つの輪郭点に対して水平方向に対をなす輪郭点、垂直方向に対をなす輪郭点の2つを抽出し、それぞれの移動量を表すフラグを取り出す。そして、ステップS1505で移動量を計算する。ステップS1505ではウエイト1からウエイト10に変化した場合の輪郭点の水平方向及び垂直方向の移動量が求められるので、これをステップS1506において各輪郭点のベクトルX座標、ベクトルY座標としてセットする。ベクトルX座標及びベクトルY座標を用いて各ウエイトに対する輪郭点の位置は実施例1と同様であるのでここでは説明を省略する。

【0046】

ステップS1504における、ある輪郭点と対になる輪郭点の抽出の方法については種々の方法が可能である。その1例を図16のフローチャートを用いて説明する。例えば、ある輪郭点を基準に水平方向及び垂直方向に走査して、輪郭点と水平方向及び垂直方向に対となる輪郭点を抽出する。

【0047】

図16は移動量を計算するサブルーチンであり、図15のステップS1505の処理の詳細を表すフローチャートである。尚、移動量は、水平移動量大／小フラグ或いは垂直移動量大／小フラグの種類によりあらかじめ定められており、対の輪郭点のフラグと標準線幅から移動量の算出を行う。

【0048】

まず、ステップS1601で、当該輪郭点の水平移動量を図14のフローチャートでセットされたフラグによりセットする。また、ステップS1503で取り出された水平方向に対をなす輪郭点の水平移動量を移動量を表すフラグによりセットする。ステップS1602では、当該輪郭点の垂直移動量を図14のフローチャートでセットされたフラグによりセットする。更に、ステップS1503で抽出された垂直方向に対をなす輪郭点の垂直移動量を移動量を表すフラグに基づいてセットする。

【0049】

そして、ステップS1603では、注目している輪郭点の水平移動量と、当該輪郭点と水平方向に対をなす輪郭点の水平移動量と、ウエイト1における標準水平線幅とを足し合わせて、これをWに代入する。同様に、ステップS1604では、注目している輪郭点の垂直移動量と、当該輪郭点と垂直方向に対をなす輪郭点の垂直移動量と、ウエイト1における標準垂直線幅とを足し合わせて、これをHに代入する。

【0050】

ステップS1605においては、ステップS1603で代入されたWの値と、ウエイト10における標準水平線幅とを比較する。ここで、Wの方が大きい場合は水平線幅が太くなり過ぎることを意味するので、ステップS1606で移動量を補正する。即ち、ステップS1606では、ステップS1601でセットされた輪郭点の水平移動量に、

(ウエイト10の標準水平線幅－ウエイト1の標準水平線幅)

／(W－ウエイト1の標準水平線幅)

を乗じることで、ウエイト10における水平線幅が標準水平線幅を越えないよう

に補正する。1方、ステップS1605でWがウエイト10の標準水平線幅以下であれば、ステップS1601でセットされた水平移動量をそのまま用いる。

#### 【0051】

ステップS1607及びステップS1608では垂直方向について上述のステップS1605及びステップS1606と同様の処理を行う。このようにして、ウエイト10における各輪郭点の移動量が算出されるので、これを用いて実施例1と同様の手法で輪郭点移動ベクトルを得ることができる。

#### 【0052】

また、図15のステップS1504における対をなす輪郭点の抽出方法としては種々の方法が考えられるが、その1例を図22及び23を用いて説明する。図22はある注目輪郭点と対をなす輪郭点を抽出するための手順を表すフローチャートであり、図23はある注目輪郭点と対をなす輪郭点を抽出するための手順を説明するための図である。

#### 【0053】

まず、ステップS2201において、注目する輪郭点を始点とする輪郭ベクトルを抽出する。同様にステップS2202において、注目する輪郭点を終点とする輪郭ベクトルを抽出する。ここで輪郭ベクトルは、図23に示されるように輪郭の内側がベクトルの方向に対して常に左側にあるように構成される。以上のステップS2201および2202において、例えば図23の輪郭点2300を始点とする輪郭ベクトル2300b及び輪郭点2300を終点とする輪郭ベクトル2300aが抽出される。

#### 【0054】

次に、ステップS2203において、上記抽出された輪郭ベクトルの方向の左側へ水平に走査し、対をなす輪郭ベクトルを検索する。図23を用いて説明すると、輪郭ベクトル2300bより水平方向に、例えば走査線2300cに沿って輪郭ベクトルを検索すると、対ベクトルとして輪郭ベクトル2305bが抽出される。また、走査線2300dに沿って輪郭ベクトルを検索すると輪郭ベクトル2302bが抽出される。

#### 【0055】



そして、ステップS2204において、抽出された対ベクトルが複数か否かをチェックし、複数でなければステップS2205へ進み、複数ある場合はステップS2204へ進む。ステップS2205では、注目する輪郭点を含む輪郭ベクトルと水平方向に最も近い位置にある対ベクトルを抽出する。これを図23を用いて説明すると、輪郭ベクトル2305bよりも輪郭ベクトル2302bの方が近いので、これを対ベクトルとする。

【0056】

ステップS2206では、以上のようにして抽出された輪郭ベクトルの両端点のうち、注目している輪郭点に近い方の端点を対をなす輪郭点とする。即ち、図23において、輪郭ベクトル2302bの両端点のうち輪郭点2300に近い方の端点は輪郭点2301であるので、これを水平方向に対をなす輪郭点とする。

【0057】

ステップS2207からステップS2210では、垂直方向に対して走査を行い垂直方向に対をなす輪郭点を求める。その内容は上記の水平方向の場合と同様であるのでここではその説明を省略する。図23において、輪郭点2300の垂直方向に対をなす輪郭点として、輪郭点2301が得られる。また、例えば輪郭点2302について上述の処理を実行すると、輪郭点2303が水平方向に対をなす輪郭点として、輪郭点2304が垂直方向に対をなす輪郭点として抽出されることがわかる。

【0058】

以上説明したように、文字輪郭上の各点にウェイトを可変にするためのベクトルを1種類のウェイトから自動計算することにより以下のような効果がある。即ち、

1. 1種類のウェイトから他のウェイトを作成するベクトルを計算できるため、文字デザインの工程が短縮される
  2. 複数の輪郭情報からベクトルを計算することに比べ、データ管理が容易である
- という効果がある。

【0059】

#### ＜実施例4＞

実施例4では、上述の実施例3と同様に1種類のウエイトの輪郭点から輪郭点移動ベクトルを求めるが、ある輪郭点の移動量フラグとその輪郭点と対をなす輪郭点の移動量フラグの組み合わせによって決定される移動量を、予めテーブルとして持たせてある輪郭形成装置について説明する。尚、実施例3の輪郭形成装置の構成は実施例1と同様であるのでここではその説明は省略する。

#### 【0060】

図17から図19は、図1の装置で用いるデータを輪郭の点の位置から自動的に求める方法を表したフローチャートである。実施例3で示した方法に比べ、移動量をテーブルで持たせることにより、計算を省力化でき、テーブルの分類を細かくすることでより品位を上げることが容易である。

#### 【0061】

図17では、文字輪郭の点の位置から、水平方向の移動に対し、水平移動不可、水平移動制限有り、水平移動制限無し of 3種類の分類を行っている。もちろん、この分類は条件を変えることで容易に細かくすることができる。この移動の分類は対になる輪郭と、他の輪郭、文字枠（ボディ枠）との位置関係から導き出すことができる。すなわち、ウエイトを10にしたときの標準水平線幅とウエイトが1の標準水平線幅の差分、輪郭点を水平に、対の輪郭とは反対方向に動かしたときに、その輪郭点の輪郭に接触もしくは交差するかどうか、また文字のボディ枠をはみ出すかどうか、といったことから移動の可能性をフラグにセットする。同様に垂直方向についても同様の判定を行う。

#### 【0062】

図17のフローチャートにおけるステップS1701からS1715の各ステップは図14のフローチャートにおけるステップS1401からS1415の各ステップと同様であり、ここではその説明は省略する。図17のフローチャートより求めた全輪郭点の移動量のフラグから、実際の移動量を求める。図18ではこの移動量を求める手続きを示している。移動量の計算では通常、輪郭の点と輪郭の対となる点のフラグとをあわせて、テーブルから移動量を取り出す。

#### 【0063】

図18における、ステップS1805を除くステップS1801から1807の各ステップは、図15におけるステップS1505を除くステップS1501から1507の各ステップと同様である。ステップS1805は、水平移動テーブル及び垂直移動テーブルを用いて移動量の絶対値を獲得するための処理であり、その詳細を図19に示す。

#### 【0064】

図19は移動量を計算するサブルーチンである。尚、ここで用いる水平移動テーブルや垂直移動テーブルは書体や文字によって変えることが可能であり、より細かなベクトルの調整が可能である。

#### 【0065】

図19について説明する。まず、ステップS1901で、注目している輪郭点の移動量を表すフラグと、当該輪郭点と水平方向に対をなす輪郭点の移動量を表すフラグを取り出す。そして、ステップS1902において、各輪郭点の移動量を表すフラグを用いて、図20に示されるような水平移動テーブルを検索し、水平方向の移動量の絶対値を獲得する。そして、対となる輪郭点から離れる方向へ移動方向をセットする。ステップS1903では、注目している輪郭点の移動量を表すフラグと、当該輪郭点と垂直方向に対をなす輪郭点の移動量を表すフラグを取り出す。そして、ステップS1904において、各輪郭点の移動量を表すフラグを用いて、図21に示されるような垂直移動テーブルを検索し、垂直方向の移動量の絶対値を獲得する。そして、対となる輪郭点から離れる方向へ移動方向をセットする。

#### 【0066】

以上のようにして、ウエイト10における輪郭点の移動量が得られるので、これを用いて輪郭点移動ベクトルを計算し、各ウエイトに対応する輪郭点の移動量を算出することができる。

#### 【0067】

図20、21は図19の移動量の検索に用いるテーブルである。図20は水平移動量用のテーブルであり、図21は垂直移動量用のテーブルである。輪郭点の移動量のフラグと、対の点の移動量のフラグとの組み合わせからテーブルを検索

し、対応する移動量を獲得し、ベクトルの生成に用いる。

【0068】

以上説明したように、文字輪郭上の各点にウェイトを可変にするための輪郭点移動ベクトルを1種類のウェイトからテーブルを用いて設定することにより、以下のような効果がある。即ち、

1. 1種類のウェイトから他のウェイトを作成するベクトルを計算できるため、文字デザインの工程が短縮される
  2. 複数の輪郭情報からベクトルを計算することに比べデータ管理が容易である
  3. テーブルを複数持つことでさまざまなフォントの文字をより高品位に展開するための輪郭点移動ベクトルを作成可能である
  4. 判定条件を細かくすることで、文字をより高品位に展開できるベクトルを作成可能である
- という効果がある。

【0069】

尚、上記実施例3、4においては、メモリ容量に余裕があれば、全文字に対して予め輪郭点の移動量を計算し、これを格納しておくことで、処理のスピードを向上することもできる。

【0070】

<実施例5>

上記各実施例では輪郭点移動ベクトルを1つの関数で表して各ウェイトに対する輪郭点の位置を決定している。本実施例では、輪郭点の移動ベクトルの関数が、あるウェイトを境に変化する場合を説明する。

【0071】

図5は輪郭点移動ベクトルを表す関数が途中で変化する例を表す。この例ではウェイト4を境にX座標の関数が変化する。即ち、ウェイトが4以上ではX座標は変化しなくなる。また、Y座標は2次以上の関数により変化している。

【0072】

図24は実施例5における輪郭データの構成を表す図である。同図において、

フラグ1からフラグ4、X、Y座標、各ベクトルX、Y座標は実施例2の図11と同様でありここでは説明を省略する。フラグ5はX方向の輪郭点の移動ベクトルの関数が増加するウェイト値を表す。また、フラグ6はY方向の輪郭点の移動ベクトルの関数が増加するウェイト値を表す。例えば図5の注目点においてはフラグ5に4が格納される。また、移動ベクトルに変化がない場合は0を格納するものとする。フラグ5もしくは6のいずれかが0以外の場合は、次の欄の各座標データの表す内容が変わってくる。例えば、フラグ5に4が格納されている場合は、X座標としてウェイトが4の時の値が格納され、ウェイト値が4以上の場合に用いるデータがフラグ3およびベクトルX座標に格納される。

## 【0073】

即ち、あるウェイト値を境に輪郭点の移動ベクトルの関数が増加する場合、輪郭データは複数の欄に渡って格納される。例えば、第1番目の欄にはウェイト1から4に対するベクトルデータが、第2番目の欄にはウェイト4から10に対するベクトルデータが格納される。更に関数の増加するウェイト値が複数有る場合はこれに応じて第3番目の欄、第4番目の欄にデータを格納していけば良い。図24の例で説明すると、aという輪郭点はX方向の移動ベクトルの関数ウェイト4で増加する。そして、ウェイト4以降は第2欄のフラグ3が0なのでX座標が増加しなくなる。一方輪郭点aのY方向の移動ベクトルの増加はない（フラグ6が0でない）。

## 【0074】

また、輪郭点bはウェイト5でX方向の移動ベクトルの関数が増加し、ウェイト3と7でY方向の移動ベクトルの関数が増加する。そして、Y方向の移動ベクトルの関数は、ウェイト1～3が2次曲線、ウェイト4～7が1次の直線となり、ウェイト8～10では増加しなくなる。

## 【0075】

図25は実施例5における輪郭点のX、Y座標の変換手順を表すフローチャートである。尚、本実施例5における文字輪郭データの出力手順は実施例2の図12のフローチャートとほぼ同様であり、座標の変換手順が異なるのみである。従って、図24に示したフローチャートは図12のステップS1208からステッ

ブS1211に相当する部分のみを表している。尚、各ベクトルX座標、ベクトルY座標に用いられているウエイト値は、関数が増加するものについては個別に記憶しておくものとする。

#### 【0076】

ステップS2401では、フラグ5が0か否かをチェックし、0であれば関数の変化がないのでステップS2402へ進み、当該輪郭点の第1欄に格納されたデータを取り出す。また、フラグ5が0でない場合は、ステップS2403において、第1欄及び以下に続く欄のフラグ5の値により、用いるべき欄を決定し、該当する欄に格納されたデータを取り出す。以上のようにして、ステップS2402もしくは2403により取り出されたデータを用いてX座標の変換処理を実行する。即ち、ステップS2404においてフラグ3をチェックし、Xベクトルが存在するか否かを判断する。Xベクトルが存在する場合はステップS2405へ進み、輪郭点のX座標をウエイト情報及び輪郭点移動ベクトルのベクトルデータから計算する。このときの輪郭点の移動の軌跡を表す計算式は、フラグ3に記憶された曲線式の次数と、ベクトルX座標1, 2…を用いて決定する。

#### 【0077】

ステップS2406からステップS2410はY座標に関する処理であり、その内容は上述のステップS2401から2405と同様でありここではその説明を省略する。

#### 【0078】

以上のようにして、本実施例5の文字輪郭形成装置によれば、輪郭点の移動ベクトルの関数があるウエイト値を境に変化する場合にも対応することができる。

#### 【0079】

#### <実施例6>

図26は実施例6の輪郭形成装置の概略制御構成を表すブロック図である。同図において図1に示したブロック図と同1の機能を有する構成には同1の参照番号を付し、ここではその説明を省略する。14'はROMであり、あるウエイトの輪郭データが格納されている。この輪郭データは一般的な輪郭データであり、他のウエイトに対応するための移動情報は含まれていない。17はハードディスク

クであり、ROM14'に格納された輪郭データを元にして生成された輪郭点移動ベクトルを含む輪郭データを格納する。即ち、本実施例6の輪郭形成装置は、ROM14'に格納されている一般的な輪郭データから輪郭点の移動ベクトルデータを算出し、この移動ベクトルデータを有する輪郭データをハードディスク17に格納するものである。

#### 【0080】

図27は、ハードディスク17における輪郭データのデータ構造を表す図である。同図において、1701はヘッダー部であり、当該輪郭データの「書体名」、「文字の並び」、「コードの範囲」、「ウエイト」、「バリエーション」が含まれている。「書体名」は書体の種別（ゴシック体、明朝体等）を表す。「文字の並び」とはASCII、JIS0208等のどの体系に準拠するものであるか表す。例えばASCIIに準拠するのであれば0x42、JIS0208に準拠するのであれば0xC2のようにコードで表されている。「コードの範囲」とは、文字の割り当てられたコードの開始番号（開始コード）と終了番号（終了コード）で表される範囲のことである。例えばJIS第1水準の各漢字は、3021（16進数）～4F7E（16進数）の範囲の番号がコードとして割り当てられているので、先頭コード：3021、終了コード：4F7Eとなる。「ウエイト」は文字の太さを表す数値である。「バリエーション」とは、例えば「長体」、「平体」、「斜体」等の文字の形状を表すものである。

#### 【0081】

1702はポインタ領域であり、各文字輪郭データの格納位置を示すポインタを格納する。ポインタは、ヘッダー部の先頭から各文字の輪郭データが格納されているアドレスへのオフセットが何バイトであるかを示しており、格納されている文字数分存在する。例えばJIS第1水準の漢字の場合、 $(4F - 30 + 1) \times (7E - 21 + 1) = 3008$ 文字分のポインタが存在する（JISでは、コード番号の下2桁が7E～21までしか使用されない）。従って「先頭コードへのポインタ」とはコード番号3021の漢字の輪郭データが格納されているアドレスを特定するためのオフセット値である。同様に、「終了コードへのポインタ」とは、コード番号4F7Eの漢字の輪郭データが格納されているアドレスを特

定するためのオフセット値である。

【0082】

1703は文字輪郭データの格納領域である。ここに格納される文字輪郭データは、ROM14'に格納された輪郭データを用いて生成された、輪郭点移動ベクトルの情報を含む輪郭データであり、その内容は実施例1から5において説明してあるので、ここではその説明を省略する。

【0083】

上述の輪郭データはハードディスク17上のファイルとして管理される。ハードディスク17上には、各種処理により作成されたデータファイル（文書処理による文書データなど）が格納されており、輪郭データのファイルであるかどうかは拡張子やハードディスク17上のディレクトリ内の情報などで管理、識別される。また、文字数が多くなった場合、1つのフォントを複数のファイルに分割することも可能である。このときはヘッダー部1701に登録されている情報（書体名、文字の並び、コード範囲、ウエイト、バリエーション）から、どのフォントのどのコードに相当するファイルであるかをCPU12が判断する。

【0084】

図28は実施例6におけるデータ出力手順を表すフローチャートである。まず、ステップS2801において、文字輪郭データファイルを特定するための情報（書体名、文字コード、ウエイト、バリエーション）を入力する。ステップS2802では、ステップS2801で入力された情報と各文字輪郭データファイルのヘッダー部1701とを比較して使用する文字輪郭データファイルを特定し、ファイルを読み出す。続いてステップS2803ではステップS2801で入力された文字コードにより、当該文字コードに対応する文字輪郭データを読み出す。次に、ステップS2804において、CPU12は指定されたウエイトに従った文字輪郭点の位置を移動ベクトルデータを用いて算出する。輪郭点の位置の発生手順は実施例1～5において説明してあるのでここではその説明を省略する。ステップS2805では、以上のようにして得られた文字輪郭データを出力装置16より出力する。

【0085】



次に、輪郭点移動ベクトルに関する情報、即ち移動情報を持たない輪郭データ（以後オリジナルの輪郭データと称する）より移動情報を備えた輪郭データを生成し、これをハードディスク17に格納する手順を説明する。図29は実施例6における、輪郭データの格納手順を表すフローチャートである。

#### 【0086】

ステップS2901において、これから作成する文字輪郭データファイルに格納する輪郭データのヘッダー情報を指定する。ヘッダー情報とは、図27のヘッダー部1701に格納される情報である。ステップS2902では、設定されたヘッダー情報によりファイル出力を行い、ヘッダー部1701を作成するとともにポインタ領域1702を確保する。ここで、ポインタ領域1702の大きさはヘッダー情報うちのコードの範囲により算出される文字数により決定される。そして、ステップS2903において変数Nに文字数を代入し、カウンタCに1を代入する。尚これら変数N及びカウンタCはRAM15上に確保された領域である。

#### 【0087】

ステップS2904では、カウンタCが変数Nより大きいかどうかを判定し、大きければ本処理を終了する。一方、大きくなければステップS2905へ進み、ROM14'よりオリジナルの輪郭データとそのウェイトを読み出す。オリジナルの輪郭データが複数種類のウェイトに対して用意されていればこれら全てを読み出す。そして、ステップS2906では、読み出された輪郭データの数に応じて輪郭点移動ベクトルデータを生成する。例えば、ステップS2905において、ウェイト1のオリジナルの輪郭データとウェイト10のオリジナルの輪郭データとが読み出されていれば、1つの輪郭点に対して2つの座標が得られる。これを元に各輪郭点のウェイトをパラメータとする1次関数のベクトルデータが得られる。同様に、3つのオリジナルの輪郭点データが読み出されれば輪郭点移動ベクトルをウェイトをパラメータとする2次関数で表すことができる。このような輪郭点移動ベクトルの算出方法は実施例1及び2において説明してあるのでここではその説明を省略する。また、1つのウェイトの輪郭データしかない場合は、実施例3及び4で説明したように、移動フラグを用いて輪郭点移動ベクトルを

求める。このようにして、輪郭点移動ベクトルに関するデータが生成される。

【0088】

ステップS2907においては、ステップS2906で生成した輪郭点移動ベクトルを含めて文字輪郭データとし、文字輪郭データ格納領域1703にファイル出力する。そして、この文字輪郭データが格納された位置を、ファイルの先頭からのオフセット値としてポインタ領域1702に登録する。尚、オフセット値の登録先のポインタはカウンタCの値により決定される。そして、ステップS2908ではカウンタCを1つカウントアップし、ステップS2904に戻って上述の処理を繰り返す。

【0089】

尚、上述のステップS2906にて生成される関数の次数を輪郭データとして登録するようにしてもよい。また、ステップS2905にて読み出される輪郭点の数はオリジナルの輪郭データとして格納されている数だけ読み出しているが、読み出す輪郭点の数を予めキーボード等より2種類、3種類というように設定するようにしてもよい。

【0090】

更に、読み出された輪郭点が各ウエイトを通じて同一の位置にある場合は、ウエイトの変化に対して輪郭点が移動しないので、移動情報は無い。この場合、移動情報がない旨をフラグにて表現するようにしてもよい（図6、図11参照）。

【0091】

以上のようにして、ハードディスク17上に、複数のウエイトに対応するための輪郭点移動ベクトルデータを有する輪郭点データが格納される。従って、本実施例6の輪郭形成装置によれば、提供されたオリジナルの輪郭データから輪郭点移動ベクトル情報を生成して、ウエイトに対応するための移動情報を有する輪郭データとしてハードディスクに格納することが可能となる。従って、オリジナルの輪郭データを提供するための記憶媒体の容量が小さくて済むという効果がある。

【0092】

以上説明したように、上記各実施例の輪郭形成装置によれば、文字輪郭上の各

点にウェイトを可変にするためのベクトル情報を持たせることが可能となり、下記のような効果がある。即ち、

1. 所望のウェイト値による文字の太さをリアルタイムに計算することができる
2. 少ないメモリ容量でウェイトを変更するための情報を持たせることが可能である
3. ウェイトが違ふ文字の輪郭を高速に計算することが可能である
4. 輪郭データにベクトルを付加するためのデータ管理が容易である
5. 輪郭データに付加するベクトルデータの作成や取り出しが容易にかつ高品位に行える
6. 輪郭データにベクトルを付加する為の特別なデザインが必要なく、従来データとの共通化が容易であり、またアウトラインフォントからビットマップフォントを生成するモジュールの共有化も容易である
7. 品位に応じて計算の複雑さを変えることができるため、さまざまな装置への応用が容易である
8. データの複雑さを、メモリ容量に応じて変えることができ、省メモリ化が可能である
9. データの複雑さを、CPUに応じて変えることができ、システムに合わせたカスタマイズが可能である
10. ウェイトを変化させるベクトルの作成方法によらず、同1の展開方法で輪郭データを生成することができる。

【0093】

なお、本発明は複数の機器から構成されるシステムに適用しても、1つの機器から成る装置に適用しても良い。また、本発明はシステム或いは装置にプログラムを供給することによって達成される場合にも適用できることは言うまでもない。

【0094】

【発明の効果】

以上説明してきたように、本発明の輪郭データ記憶方法及び装置によれば、1

種類のウエイトの輪郭点データで複数のウエイトの文字を発生させることを可能とするための輪郭点の移動情報を生成し、輪郭データの各輪郭点の位置情報とともに移動情報を記憶することが可能となる。

【0095】

【図面の簡単な説明】

【図1】

本実施例の輪郭形成装置の概略制御構成を表すブロック図である。

【図2】

本実施例の輪郭形成装置をレーザビームプリンタに組み込んだ場合の概略制御構成を表すブロック図である。

【図3】

フォントデータの形式の1例を表す図である。

【図4】

図3のフォントデータをウエイト10で出力した状態を表す図である。

【図5】

輪郭点移動ベクトルを表す関数が途中で変化する例を表す図である。

【図6】

図3の「田」という文字のアウトラインデータの構成の1部を表す図である。

【図7】

実施例1の輪郭形成制御の手順を表すフローチャートである。

【図8】

輪郭点移動ベクトルを表す1次関数の式の求め方を示す図である。

【図9】

2つのウエイトの輪郭データから輪郭点移動ベクトルを取り出す例を表す図である。

【図10】

曲線で表される輪郭点移動ベクトルの1例を表す図である。

【図11】

実施例2における輪郭データの構成を表す図である。

【図12】

図11の形式の輪郭データから所望のウエイトの文字輪郭データを計算し出力する手順を表すフローチャートである。

【図13】

3つ以上のウエイトの輪郭データから非直線のベクトルを取り出す様子を表す図である。

【図14】

実施例3の輪郭形成装置において、各輪郭点に対する輪郭点移動ベクトルを求める手順を表すフローチャートである。

【図15】

実施例3の輪郭形成装置において、各輪郭点に対する輪郭点移動ベクトルを求める手順を表すフローチャートである。

【図16】

実施例3の輪郭形成装置において、各輪郭点に対する輪郭点移動ベクトルを求める手順を表すフローチャートである。

【図17】

実施例4の輪郭形成装置において、各輪郭点に対する輪郭点移動ベクトルを求める手順を表すフローチャートである。

【図18】

実施例4の輪郭形成装置において、各輪郭点に対する輪郭点移動ベクトルを求める手順を表すフローチャートである。

【図19】

実施例4の輪郭形成装置において、各輪郭点に対する輪郭点移動ベクトルを求める手順を表すフローチャートである。

【図20】

水平移動テーブルのデータ構成を表す図である。

【図21】

垂直移動テーブルのデータ構成を表す図である。

【図22】

注目輪郭点と対をなす輪郭点を抽出する手順を表すフローチャートである。

【図23】

注目輪郭点と対をなす輪郭点を抽出する手順を説明するための図である。

【図24】

実施例5における輪郭データの構成を表す図である。

【図25】

実施例5における輪郭点のX、Y座標の変換手順を表すフローチャートである。

【図26】

実施例6の輪郭形成装置の概略制御構成を表すブロック図である。

【図27】

ハードディスクにおける輪郭データのデータ構造を表す図である。

【図28】

実施例6におけるデータ出力手順を表すフローチャートである。

【図29】

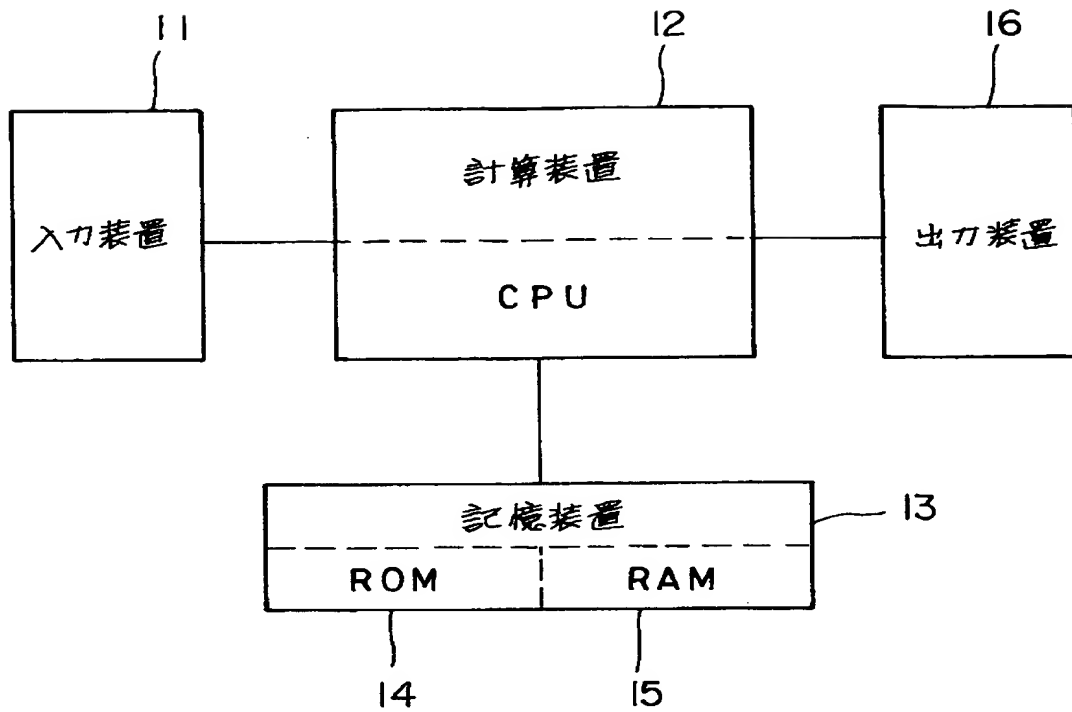
実施例6における輪郭データの格納手順を表すフローチャートである。

【符号の説明】

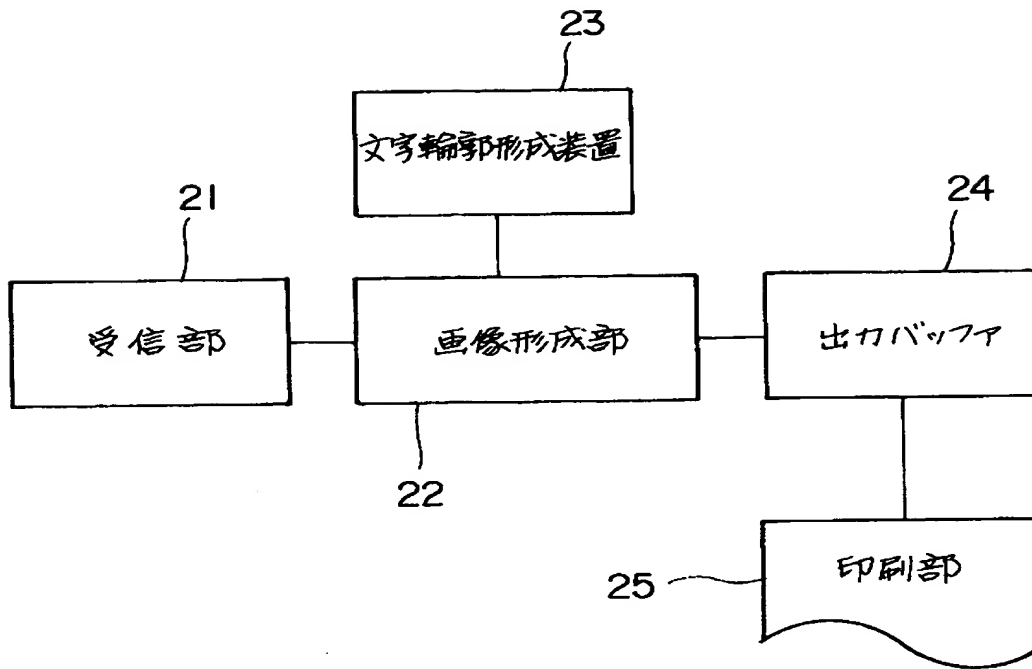
- 11 入力装置
- 12 計算装置 (CPU)
- 13 記憶装置
- 14 ROM
- 15 RAM
- 16 出力装置

【書類名】 図面

【図1】

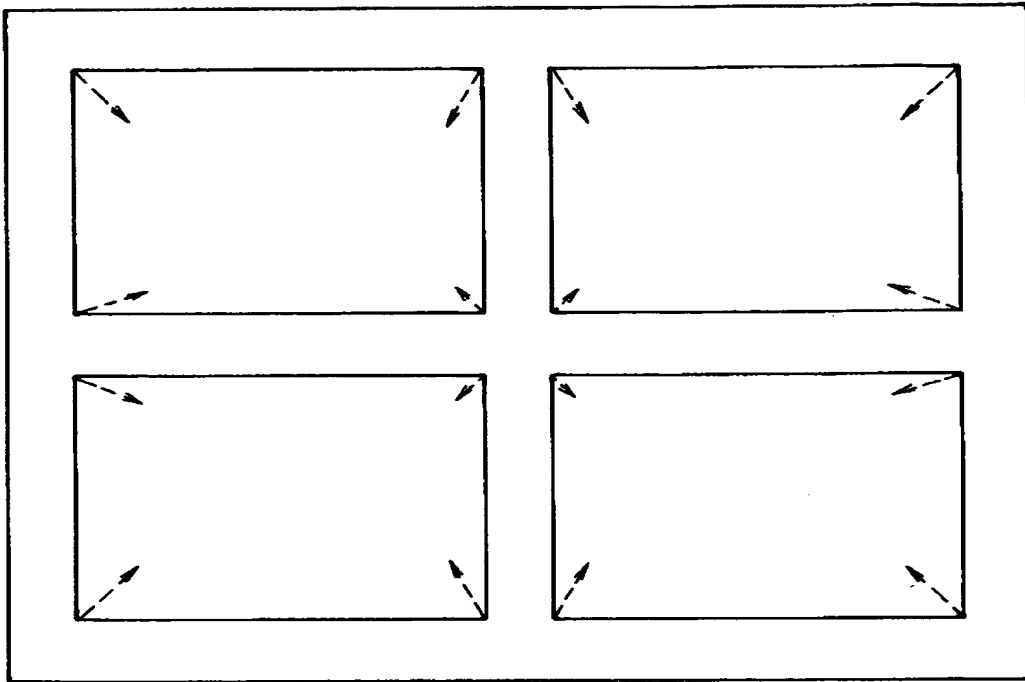


【図2】

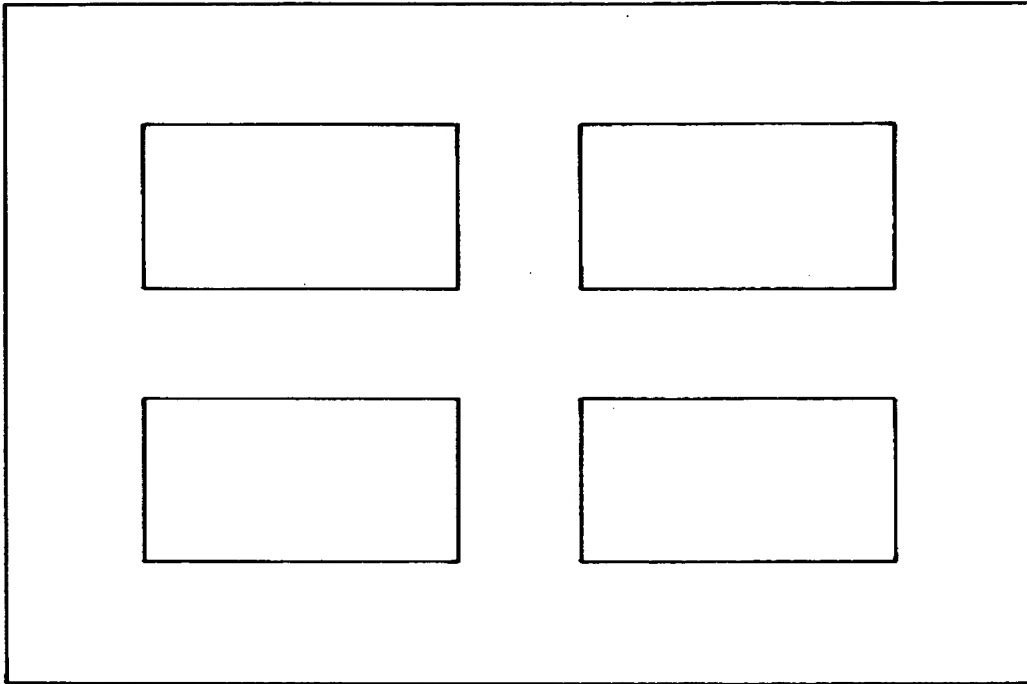




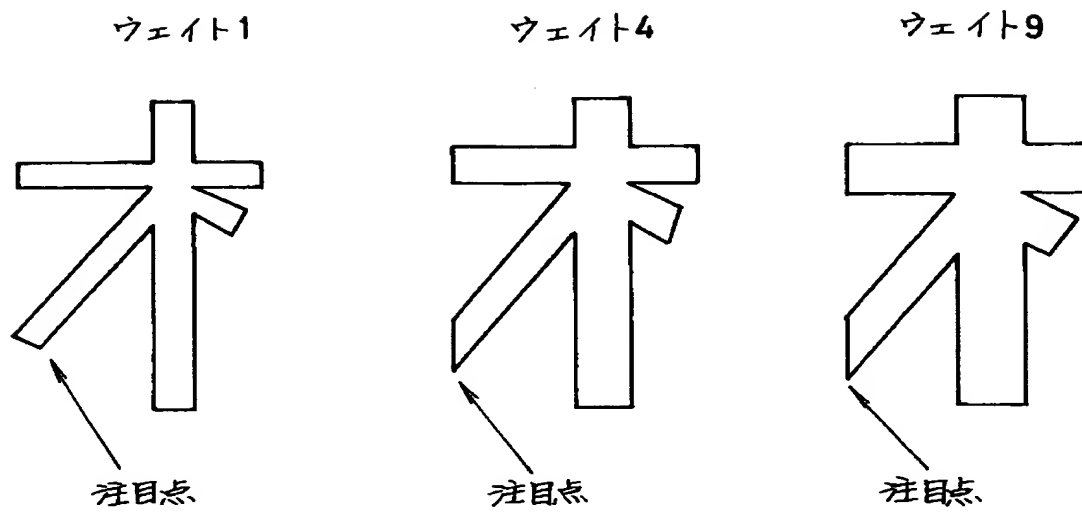
【図3】



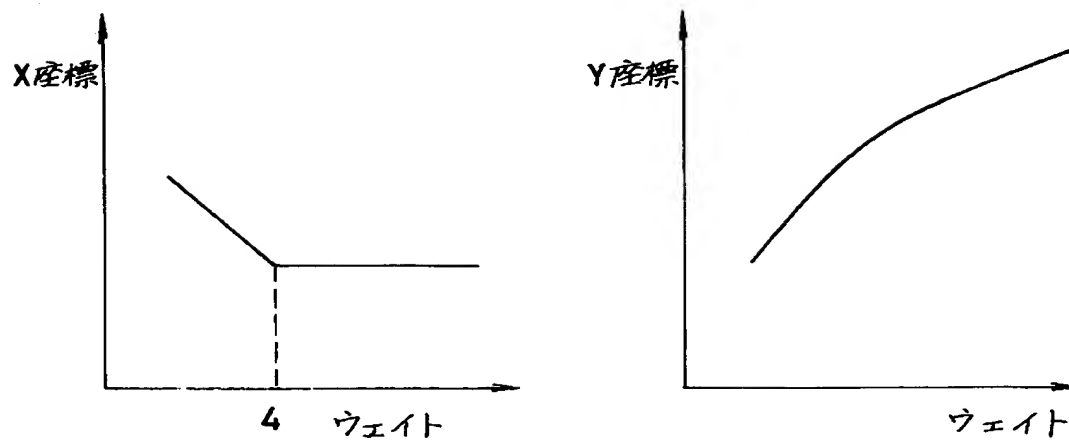
【図4】



【図5】



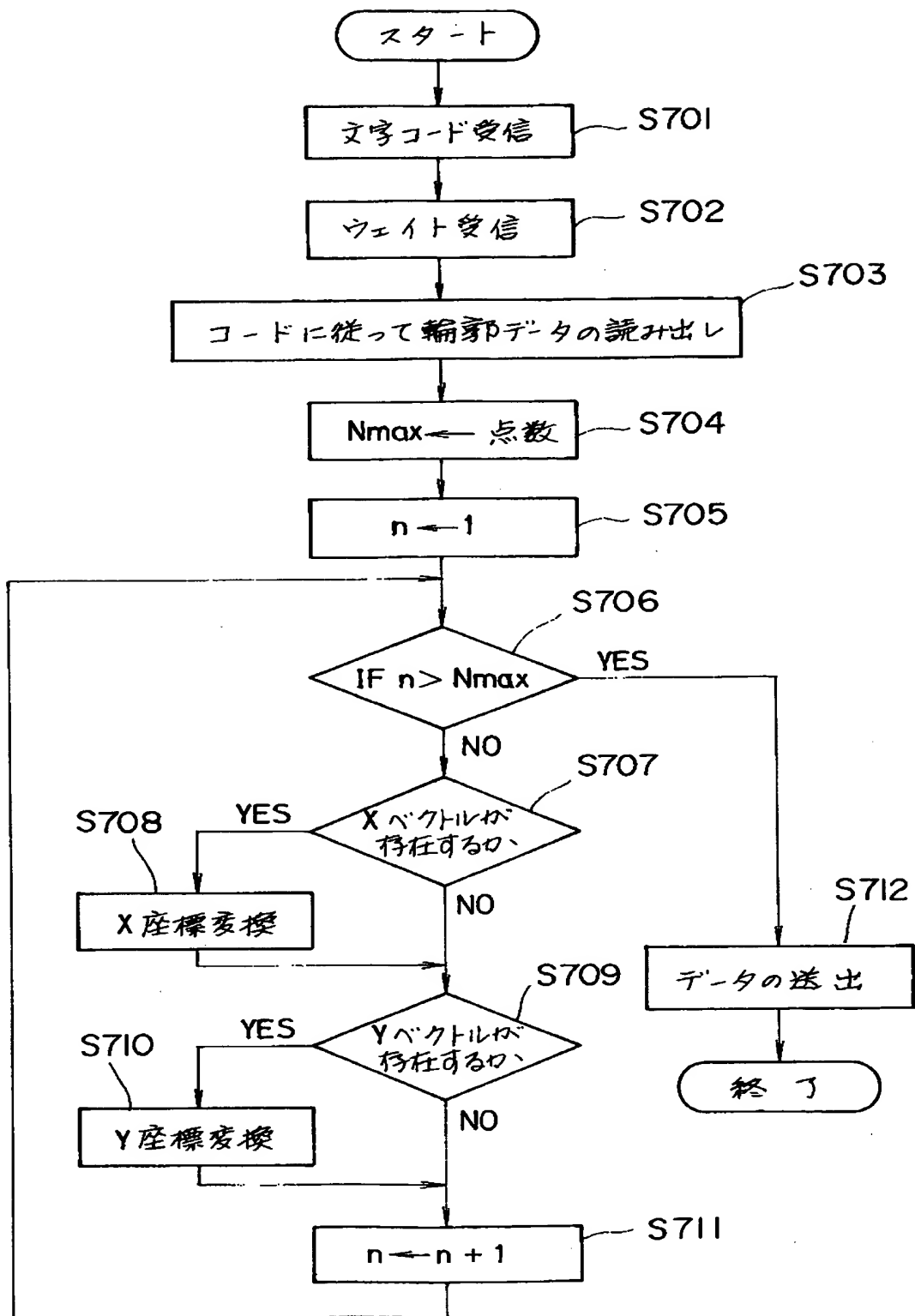
注目点のベクトル



【図6】

フラグ1	フラグ2	フラグ3	フラグ4	X座標	Y座標	バクトルX座標	バクトルY座標
開始	直線	0	0	50	50		
	直線	0	0	800	50		
	直線	0	0	800	600		
終了	直線	0	0	50	600		
開始	直線	1	1	100	100	50	50
	直線	1	1	100	300	50	-20
	直線	1	1	400	300	-20	-20
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

【図7】



【図8】

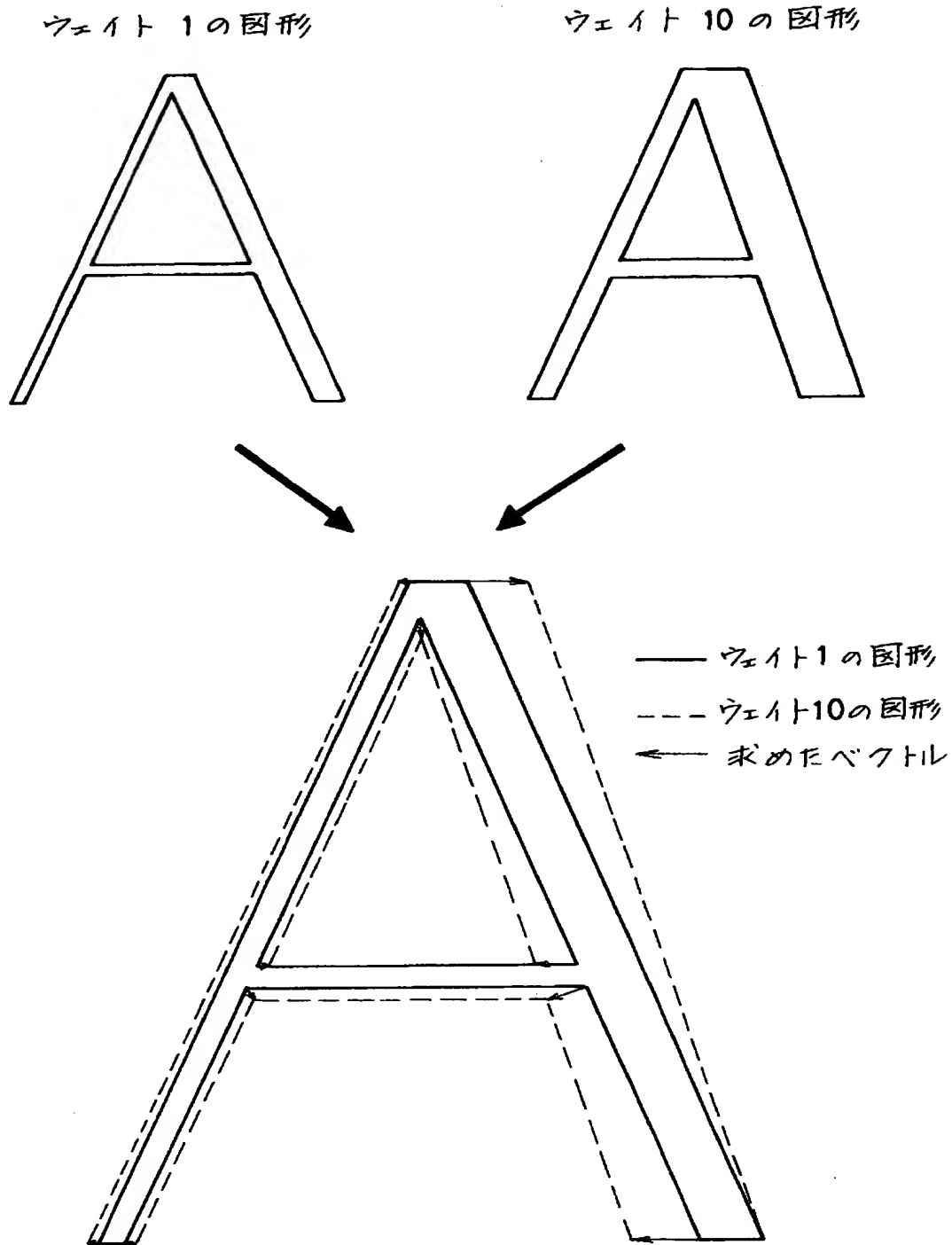
$$X = A \times W + B$$

1. Xに輪郭点のX座標、Wに1(ウェイト)を代入
2. Xにベクトルの終点のX座標、Wに10(ウェイト)を代入

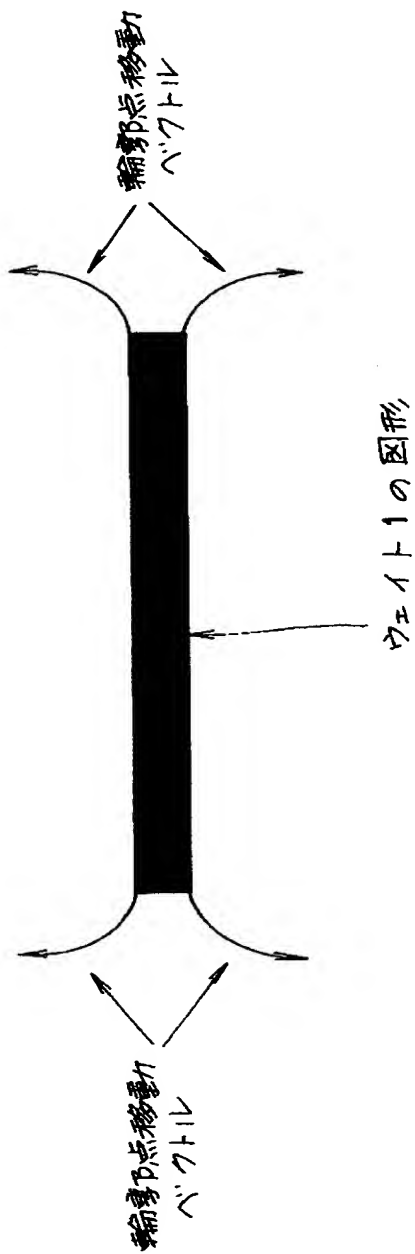
1、2よりA、Bを求めることで、任意のWについてのXが求められる

Y座標についても同様に係数を求めることで  
任意のWについてのYが求められる。

【図9】



【図10】

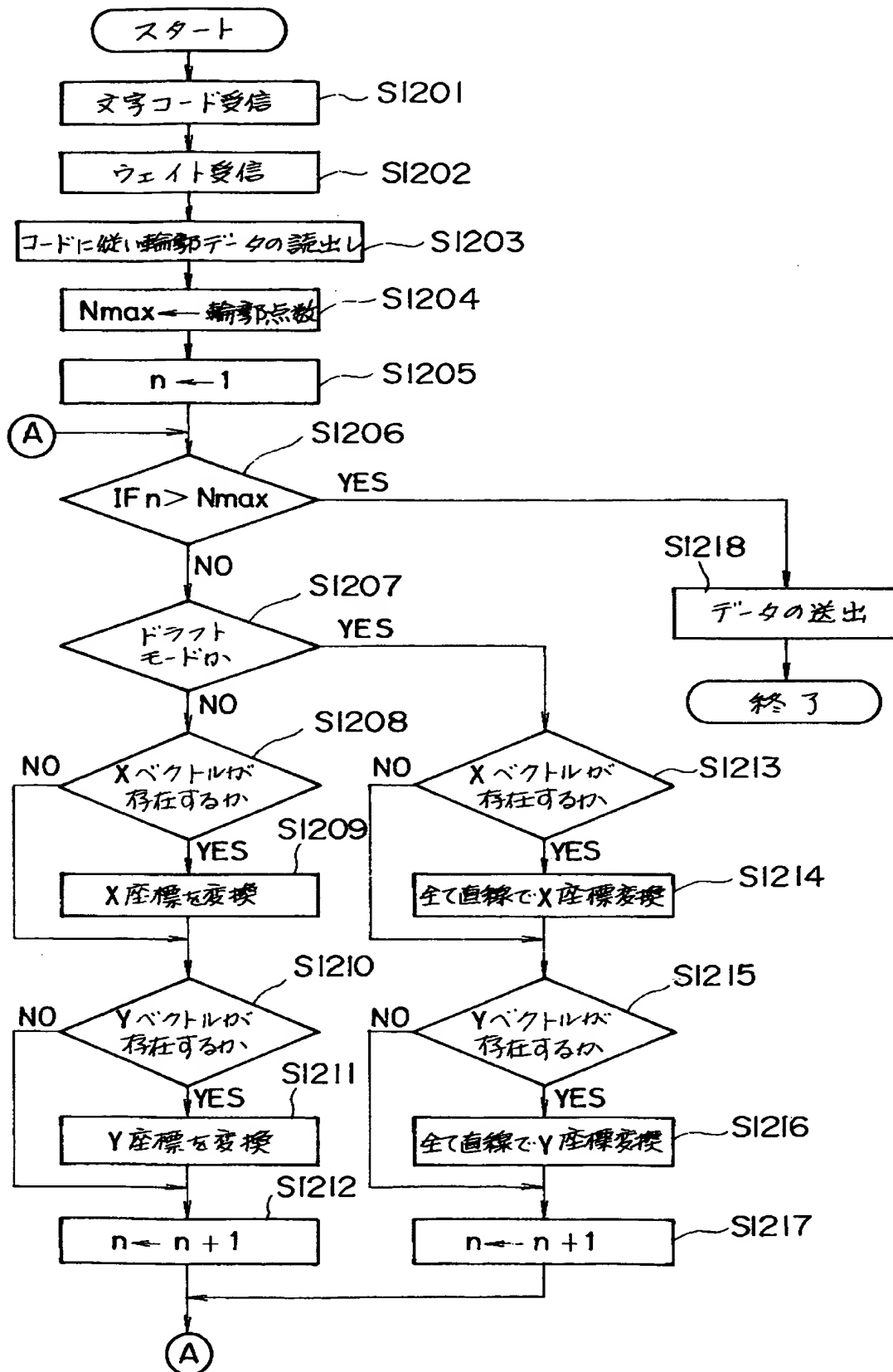




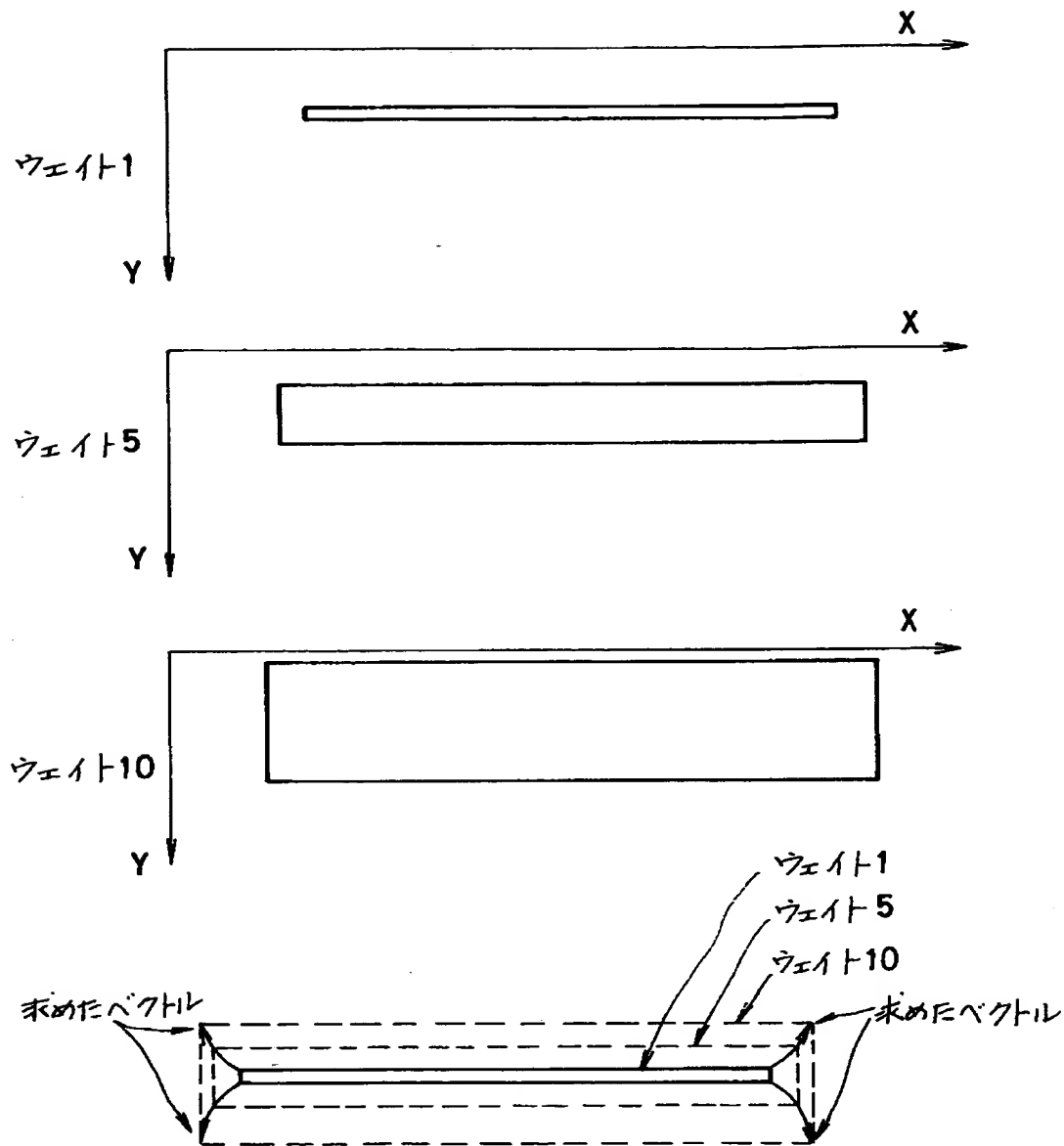
【図 1 1】

フラグ1	フラグ2	フラグ3	フラグ4	X座標	Y座標	ベクトル X座標1	ベクトル Y座標1	ベクトル X座標2	ベクトル Y座標2	ベクトル X座標3	ベクトル Y座標3
開始	直線	0	1	100	130			-40			
	曲線 開始	1	1	343	100	-10	-15				
	曲線 中間点	2	2	511	100	-20	28	-5	10.3		
	曲線 中間点	3	2	511	410	-22	-43	14	-10	-2.5	
	曲線 終了	1	1	343	410	-10	16				
終了	直線	0	1	100	380		40				
開始	直線	1	1	130	200	14	31				
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.

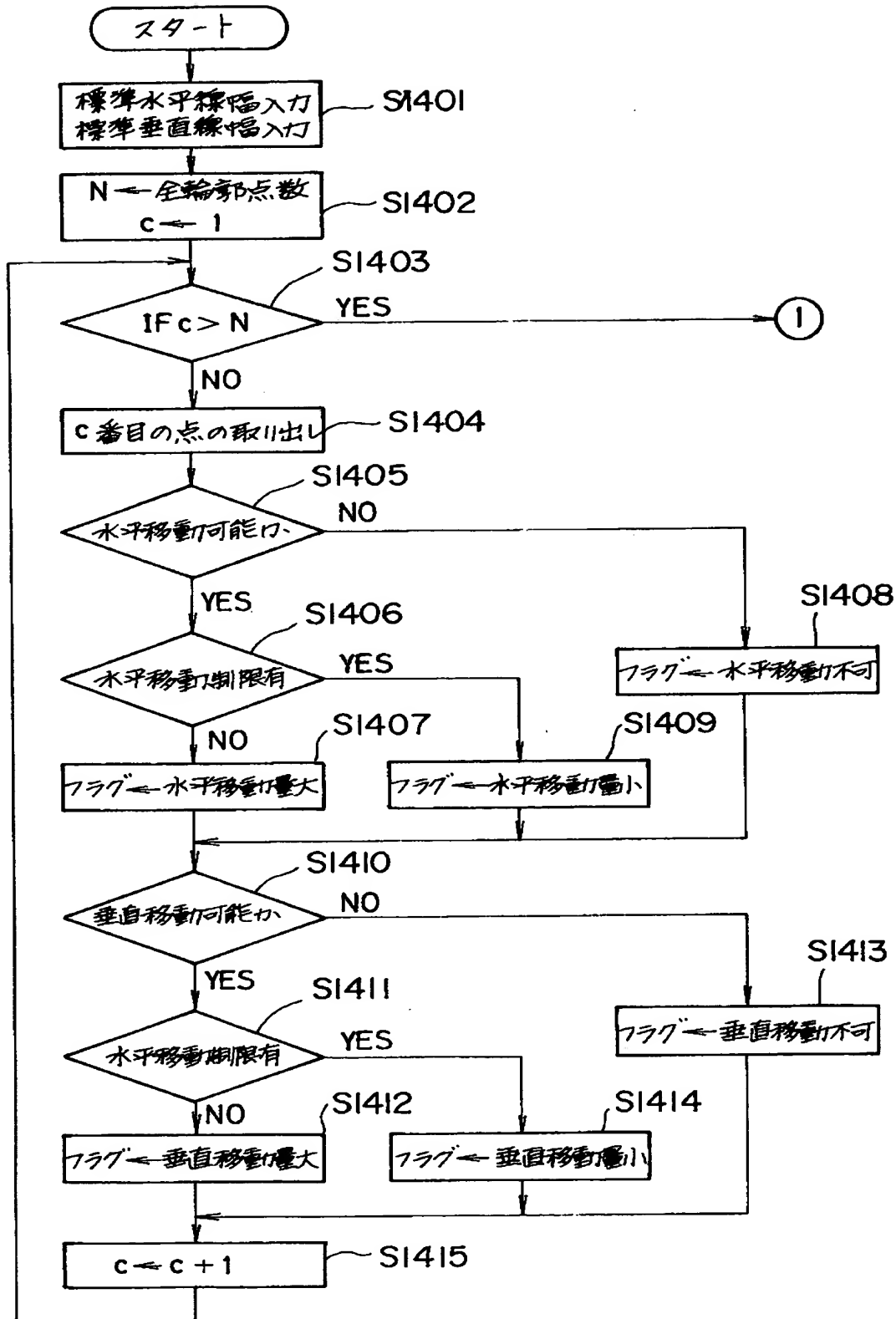
【図 1 2】



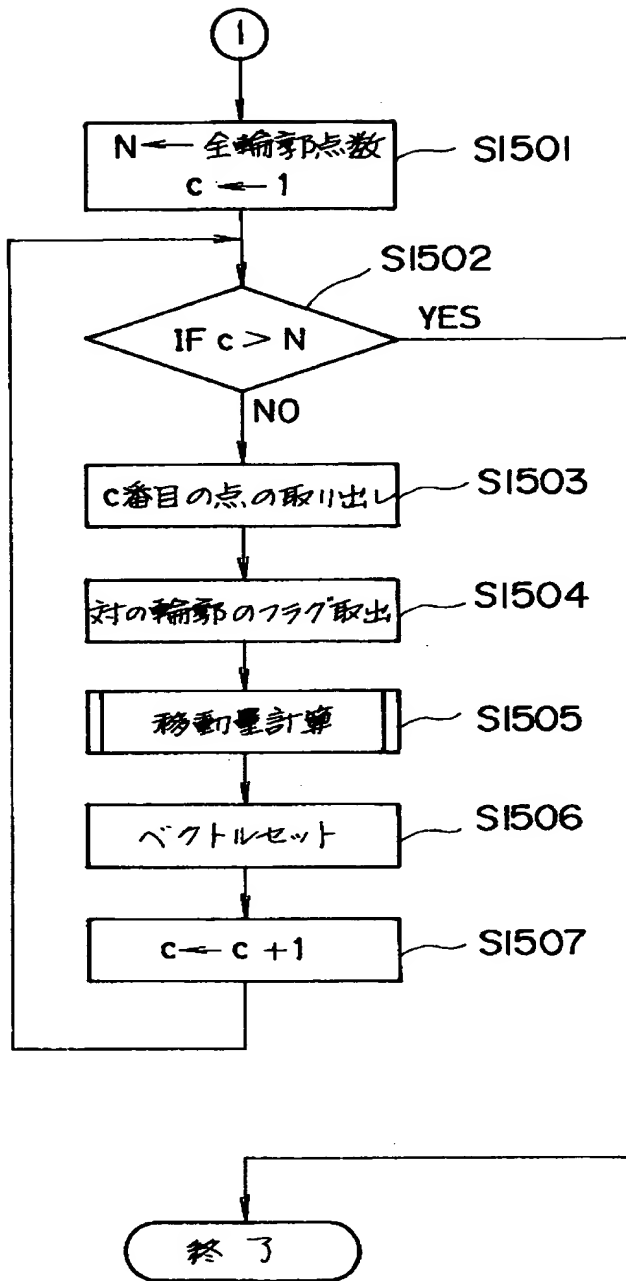
【図13】



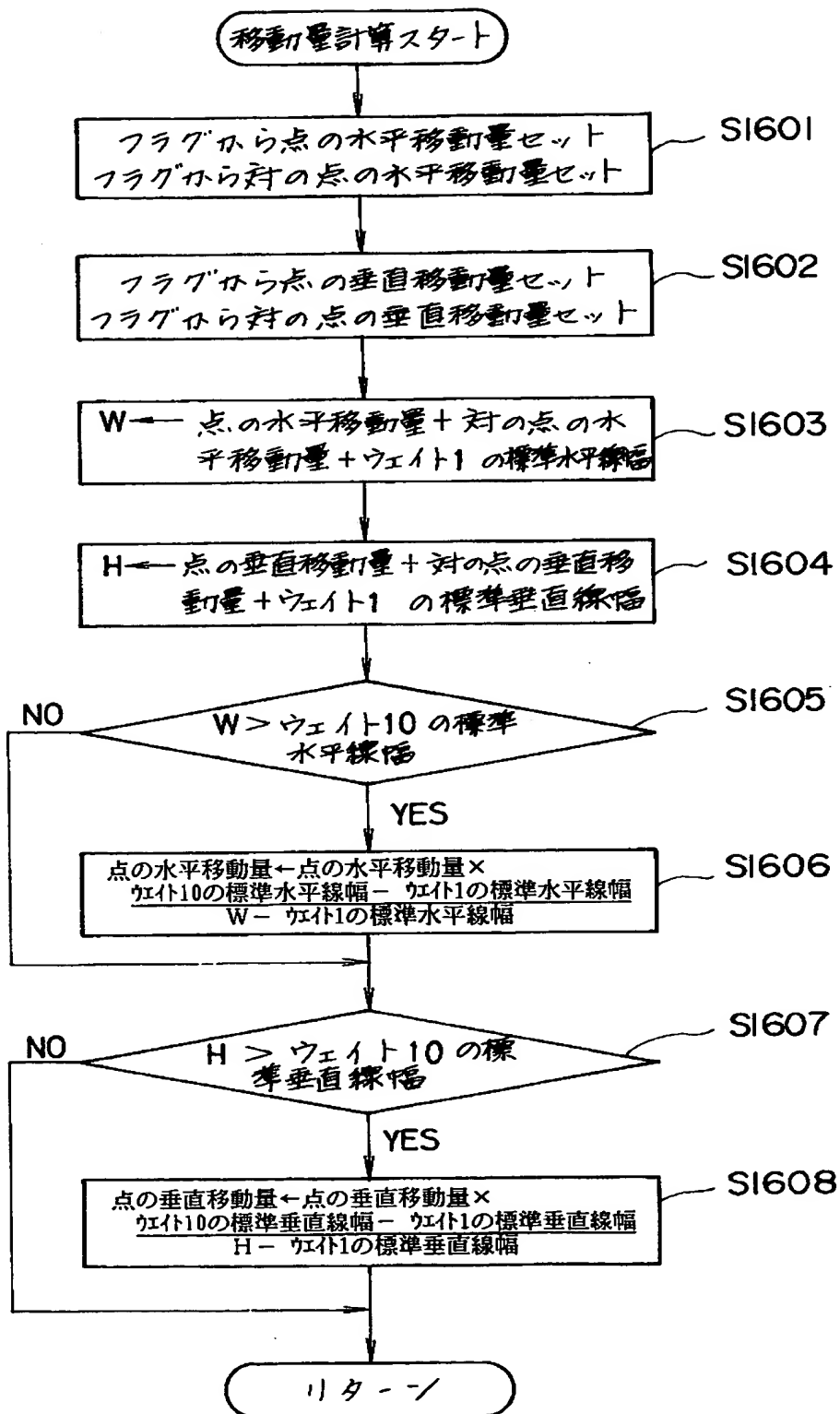
【図14】



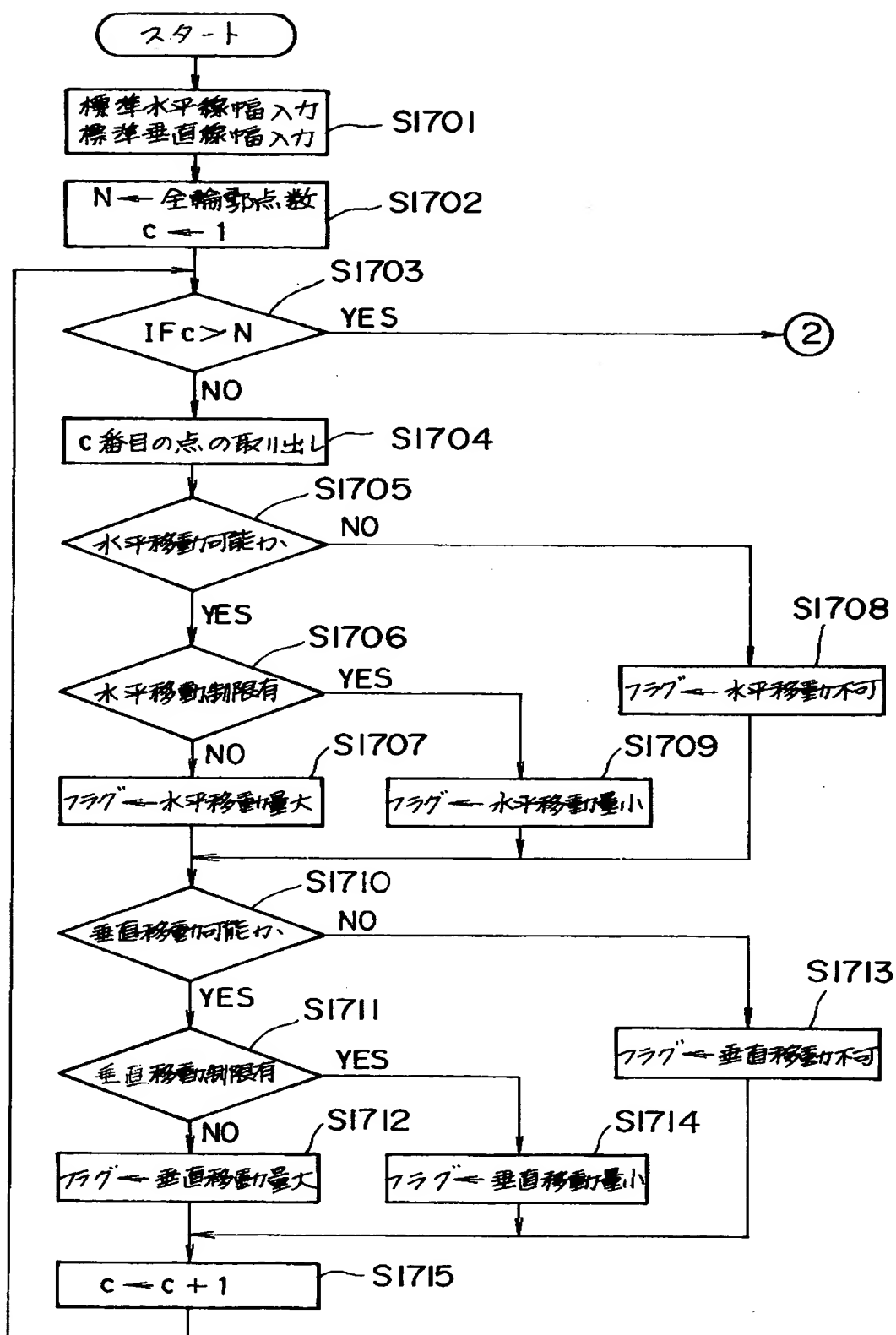
【図15】



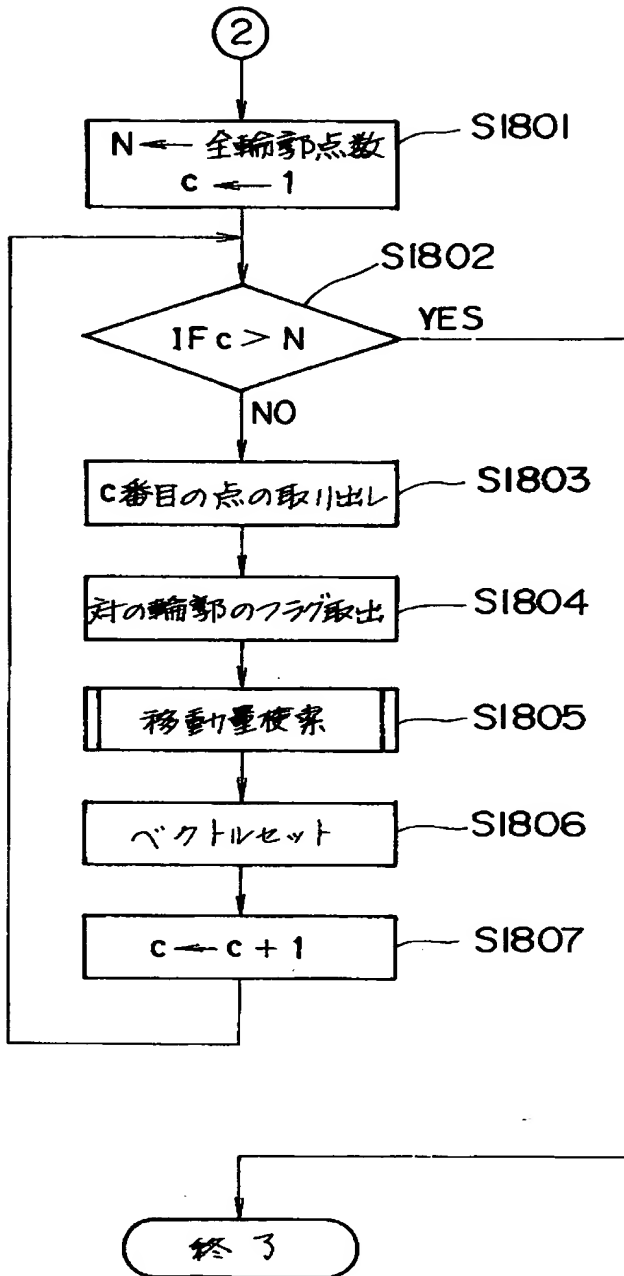
【図16】



【図17】

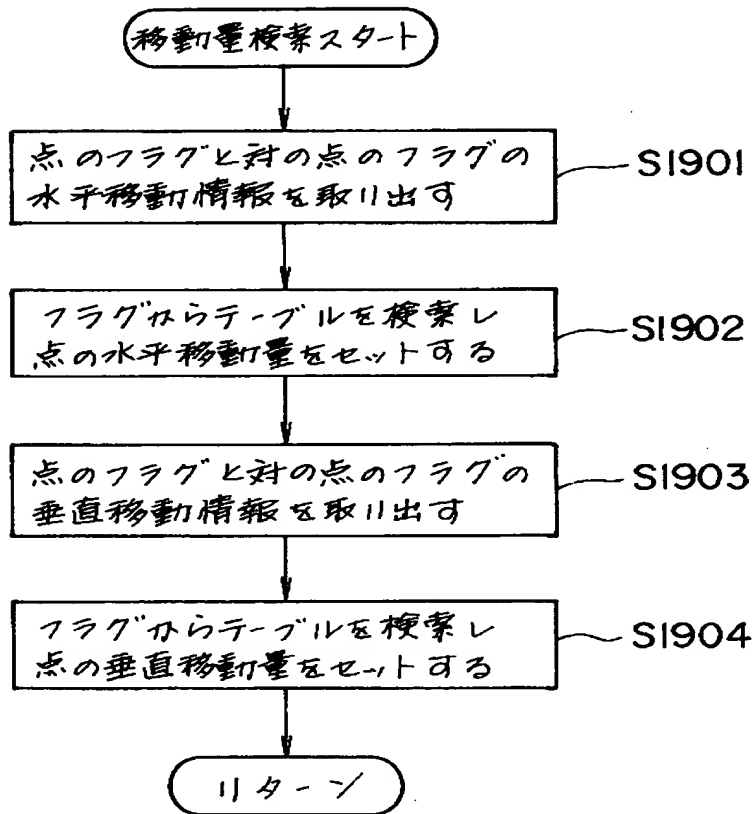


【図18】





【図19】



【図20】

水平移動テーブル

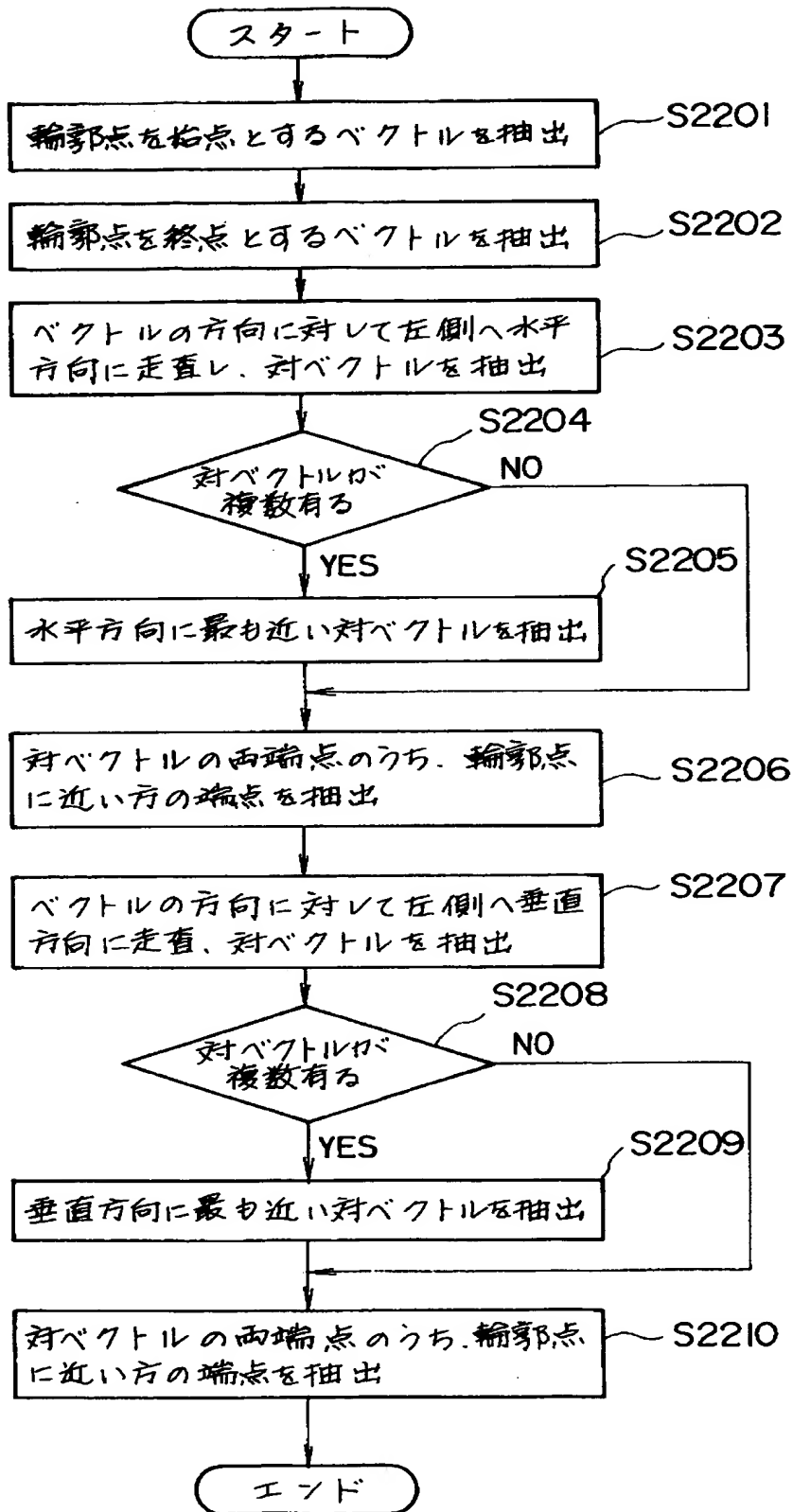
点のフラグ	対の点のフラグ	移動量の絶対値
移動不可	移動不可	0
移動不可	移動量小	0
移動不可	移動量大	0
移動量小	移動不可	10
移動量小	移動量小	16
移動量小	移動量大	7
移動量大	移動不可	20
移動量大	移動量小	13
移動量大	移動量大	10

【図 2 1】

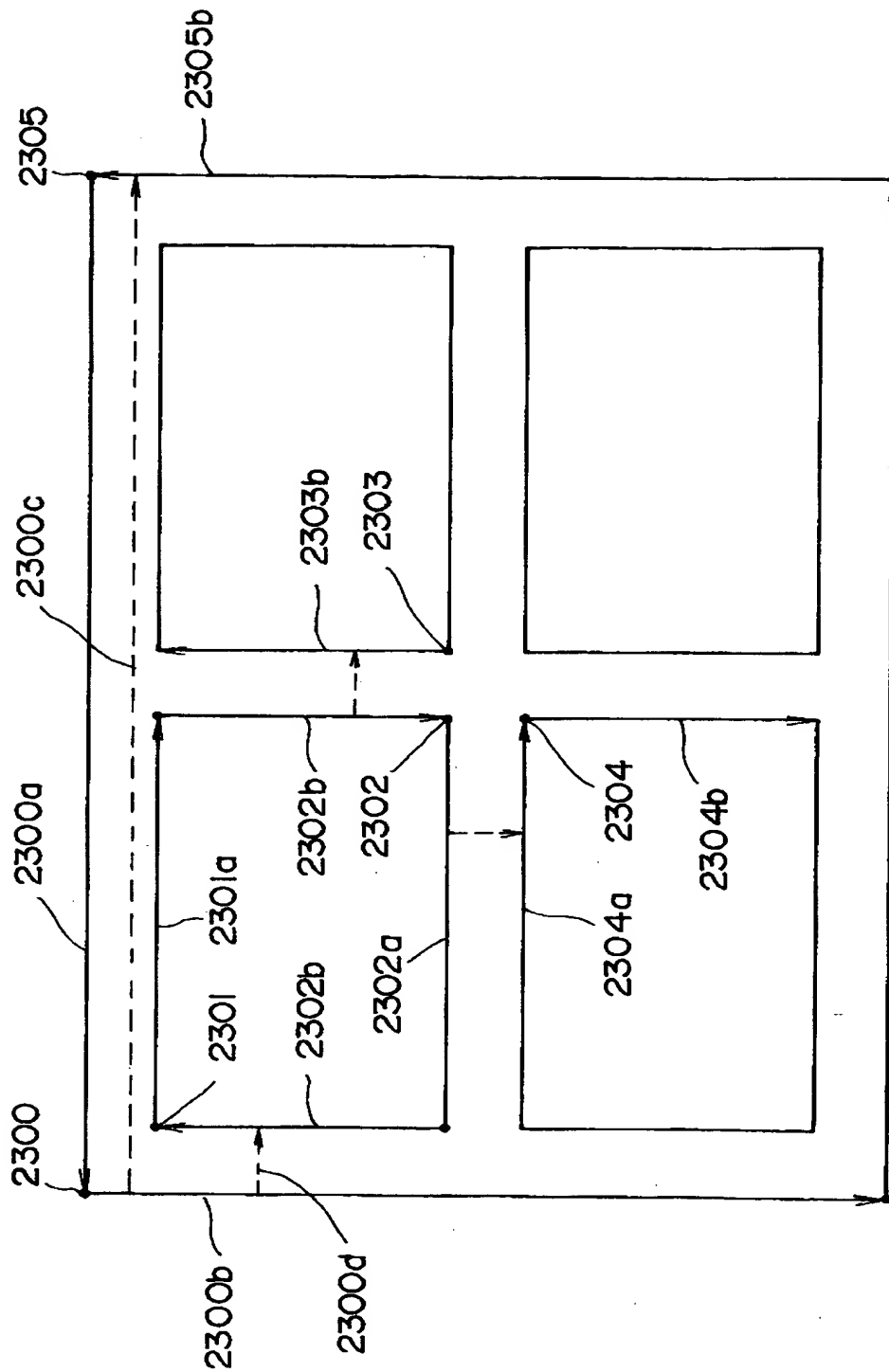
垂直移動量テーブル

点のフラグ	対の点のフラグ	移動量の絶対値
移動不可	移動不可	0
移動不可	移動量小	0
移動不可	移動量大	0
移動量小	移動不可	7
移動量小	移動量小	13
移動量小	移動量大	5
移動量大	移動不可	14
移動量大	移動量小	10
移動量大	移動量大	7

【図22】



【図 2 3】



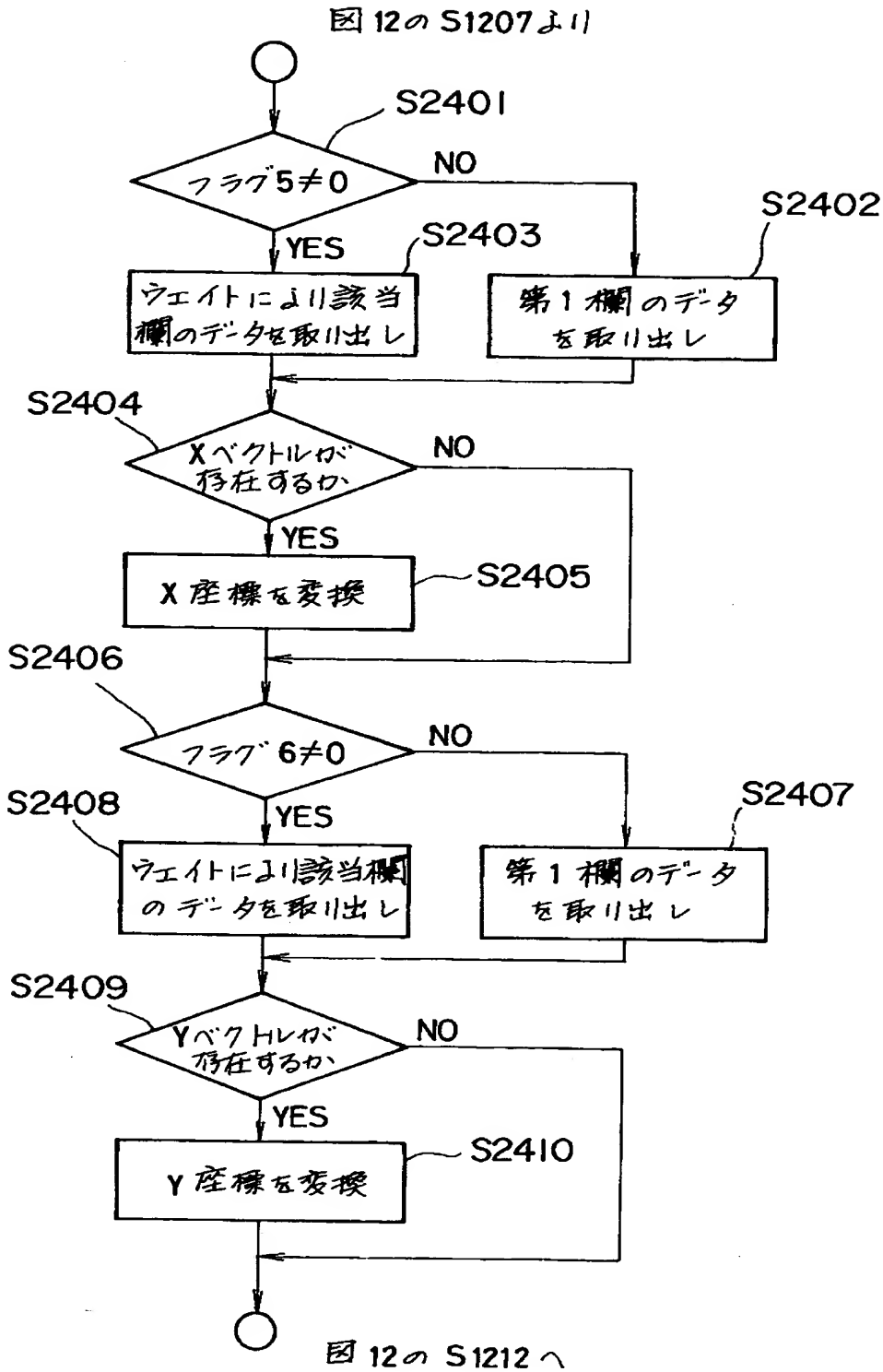
【図 2 4】

7771	7772	7773	7774	7775	7776	Y座標	バクトル X座標1	バクトル Y座標1	バクトル X座標2	バクトル Y座標2
	直線	1	2	4	0	100	130	12	-40	-10
		0		0		112				
	直線	1	2	5	3	200	400	-25	20	10
		1	1		7	175	420	-10	15	
			0		0		435			

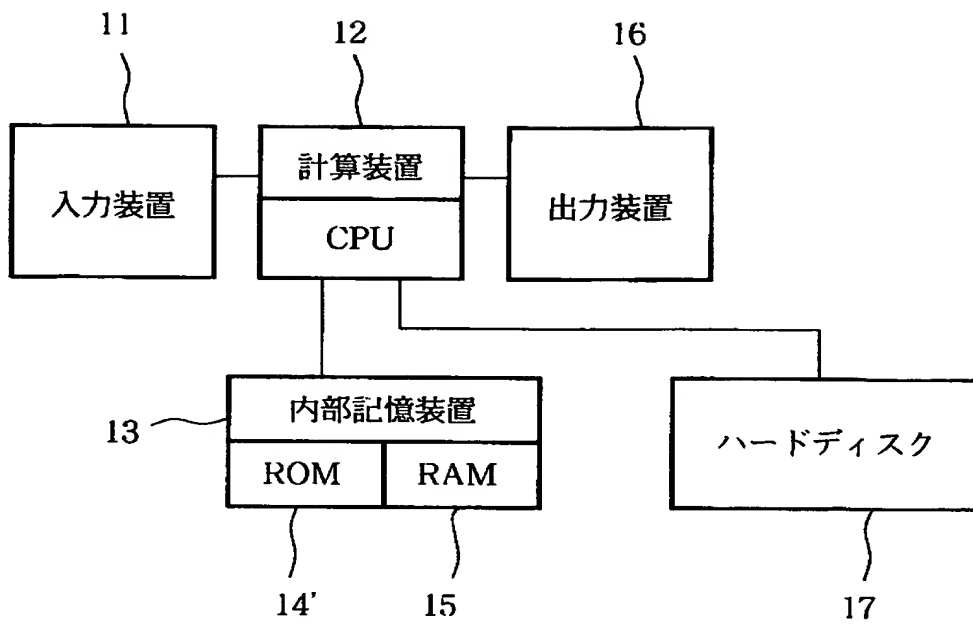
輪郭点 a

輪郭点 b

【図25】

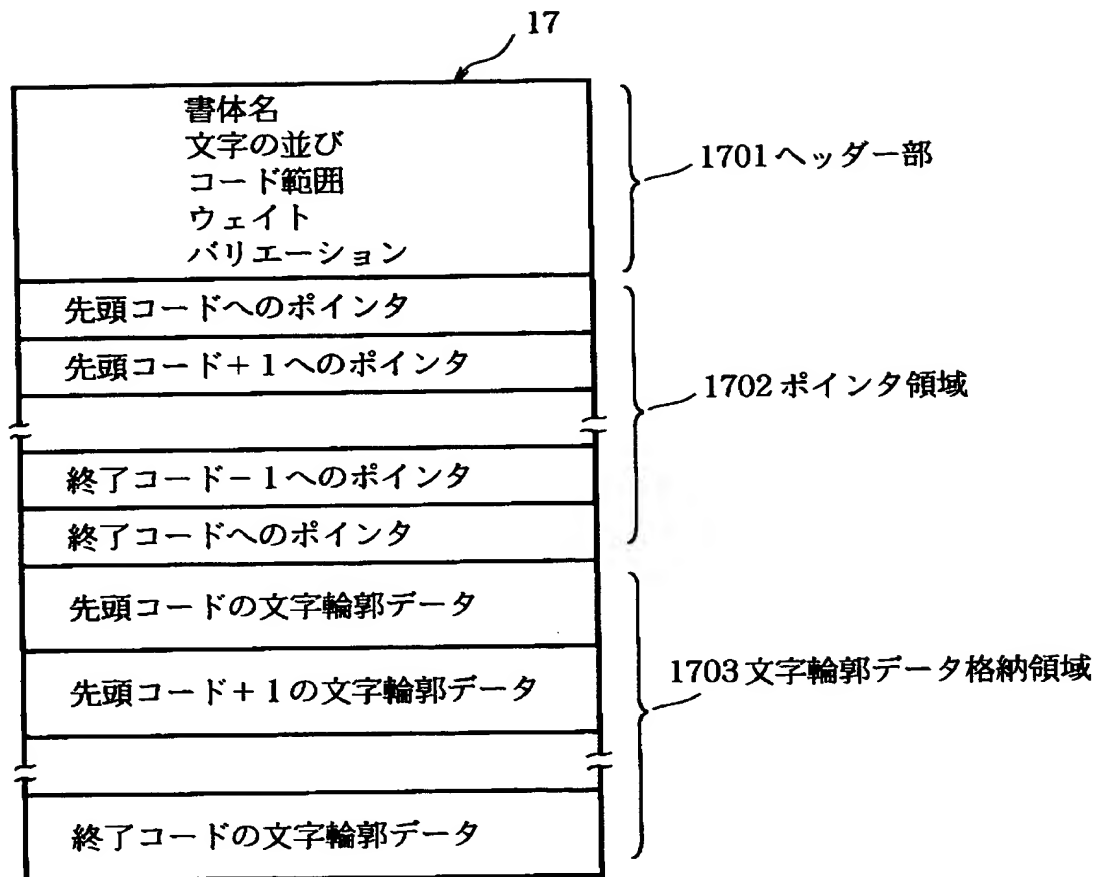


【図26】

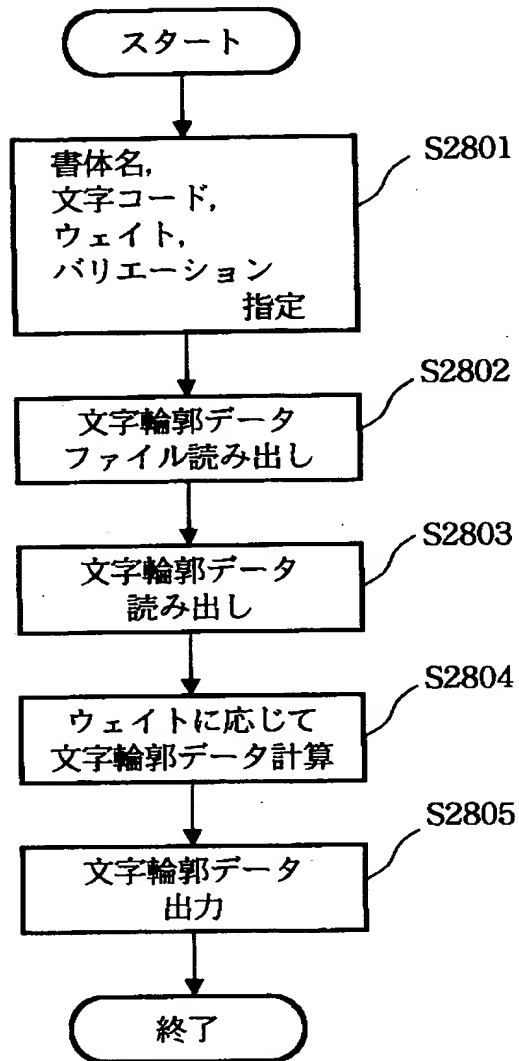




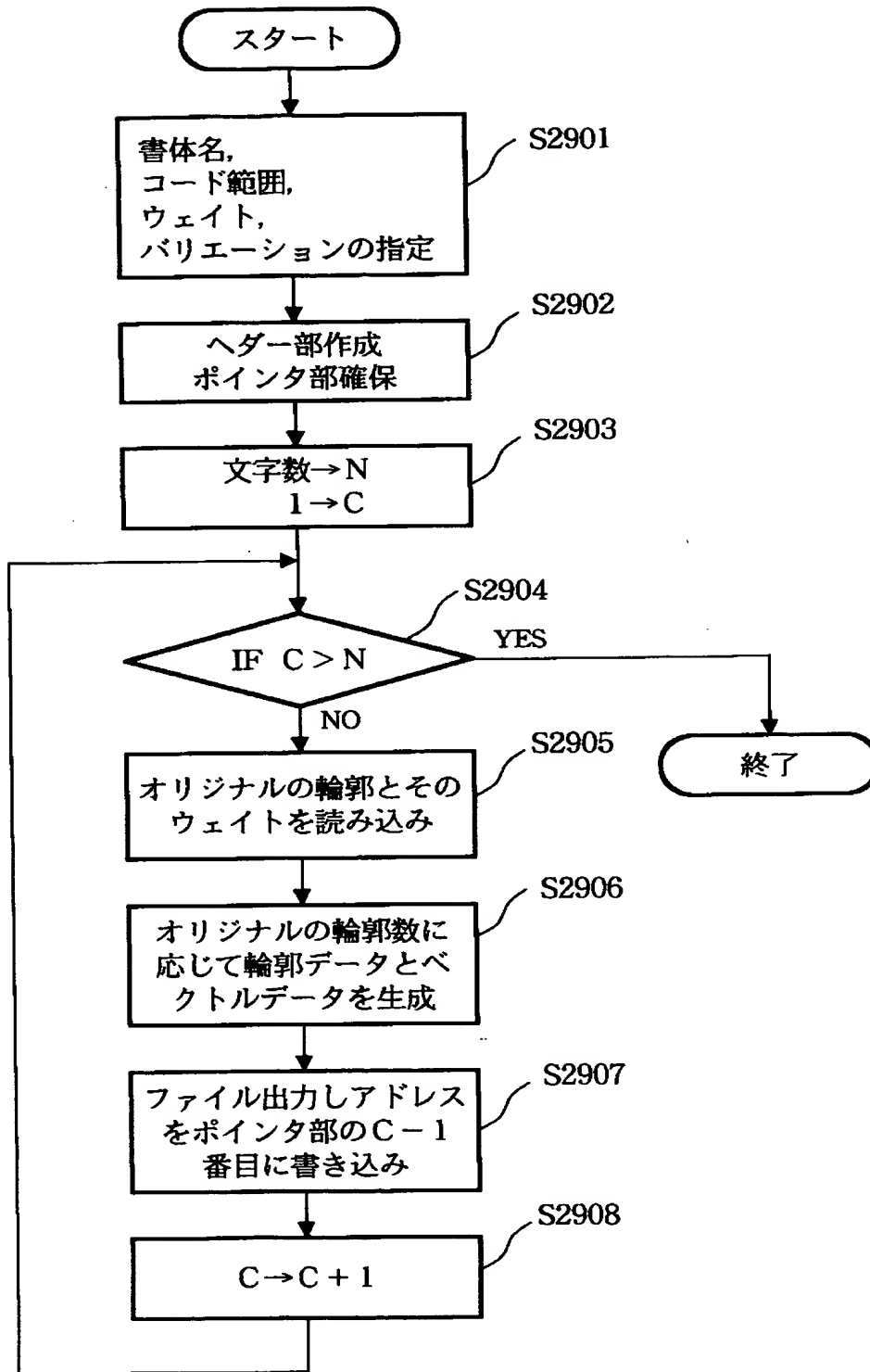
【図 27】



【図28】



【図29】



【書類名】 要約書

【要約】

【目的】 1種類のウエイトの輪郭点データで複数のウエイトの文字を発生させることを可能とするための各輪郭点の移動情報を生成し、輪郭データの各輪郭点の位置情報とともに各輪郭点の移動情報を記憶する。

【構成】 記憶媒体上に記憶されている各輪郭点の位置情報とウエイト情報を有する輪郭データを読み出す。例えばウエイト1とウエイト10の輪郭データを読み出すことにより、ウエイト1からウエイト10へ到る各輪郭点の移動の軌跡が、ウエイトをパラメータとする1次関数として得られる。このようにして得られた1次関数を生成するための情報を各輪郭点の位置情報とともに記憶媒体に記憶する。

【選択図】 図9

【書類名】 職権訂正データ  
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000001007  
【住所又は居所】 東京都大田区下丸子3丁目30番2号  
【氏名又は名称】 キヤノン株式会社

【代理人】 申請人

【識別番号】 100076428  
【住所又は居所】 東京都千代田区麴町5丁目7番地 紀尾井町TBR  
ビル507号室  
【氏名又は名称】 大塚 康德

【代理人】

【識別番号】 100093908  
【住所又は居所】 東京都千代田区麴町5丁目7番地 紀尾井町TBR  
ビル507号室  
【氏名又は名称】 松本 研一

出 願 人 履 歴 情 報

識別番号 [000001007]

1. 変更年月日 1990年 8月30日  
[変更理由] 新規登録  
住 所 東京都大田区下丸子3丁目30番2号  
氏 名 キヤノン株式会社